

DA 4000

CYBER HARDWARE FOR ANALYSTS

Section 1 of 2

## CYBER HARDWARE FOR ANALYSTS

**DESCRIPTION:** This course provides to the software analyst a detailed explanation of the functioning of the Cyber 70 and 170 series mainframes. Upon completion of this course the student will be able to program the CPU and PPU more efficiently, analyse and solve system problems more effectively and to better communicate with the Customer Engineer.

**PRE-REQUISTES:** CP Compass, PP Compass

COURSE CONTENT

## HARDWARE CONFIGURATION

- Detailed block diagram
- Signal and data flow

## COMPUTER MATHEMATICS

- Complement arithmetic
- Boolean algebra
- Floating point

## HARDWARE COMPONENTS

- Logic symbology
- Registers, adders, function translators, shifters.
- Core & MOS memory operation

## PERIPHERAL PROCESSOR ORGANIZATION

- Barrel & slot concept
- Storage sequence control
- Trip counter, barrel & slot paths
- PPU instructions, flow charts

## PERIPHERAL EQUIPMENT CONTROLLER

- 3000 Series controllers
- 6000 Series controllers
- Buffer Controller types
- Processing of connect, function status, interrupts, faults, data.

## CHANNEL OPERATION

- Control signal and data flow
- Data channel converter
- Instruction flowcharts
- Status & control
- Deadstart

## CENTRAL MEMORY

- Organization, addressing, phasing
- Timing access priorities
- PPU to CM communication
- Exchange jump

## CENTRAL PROCESSOR

- Organization
- Instruction Control timing
- Memory referencing instructions
- Floating point operation
- Compare move operation
- Other types of instructions

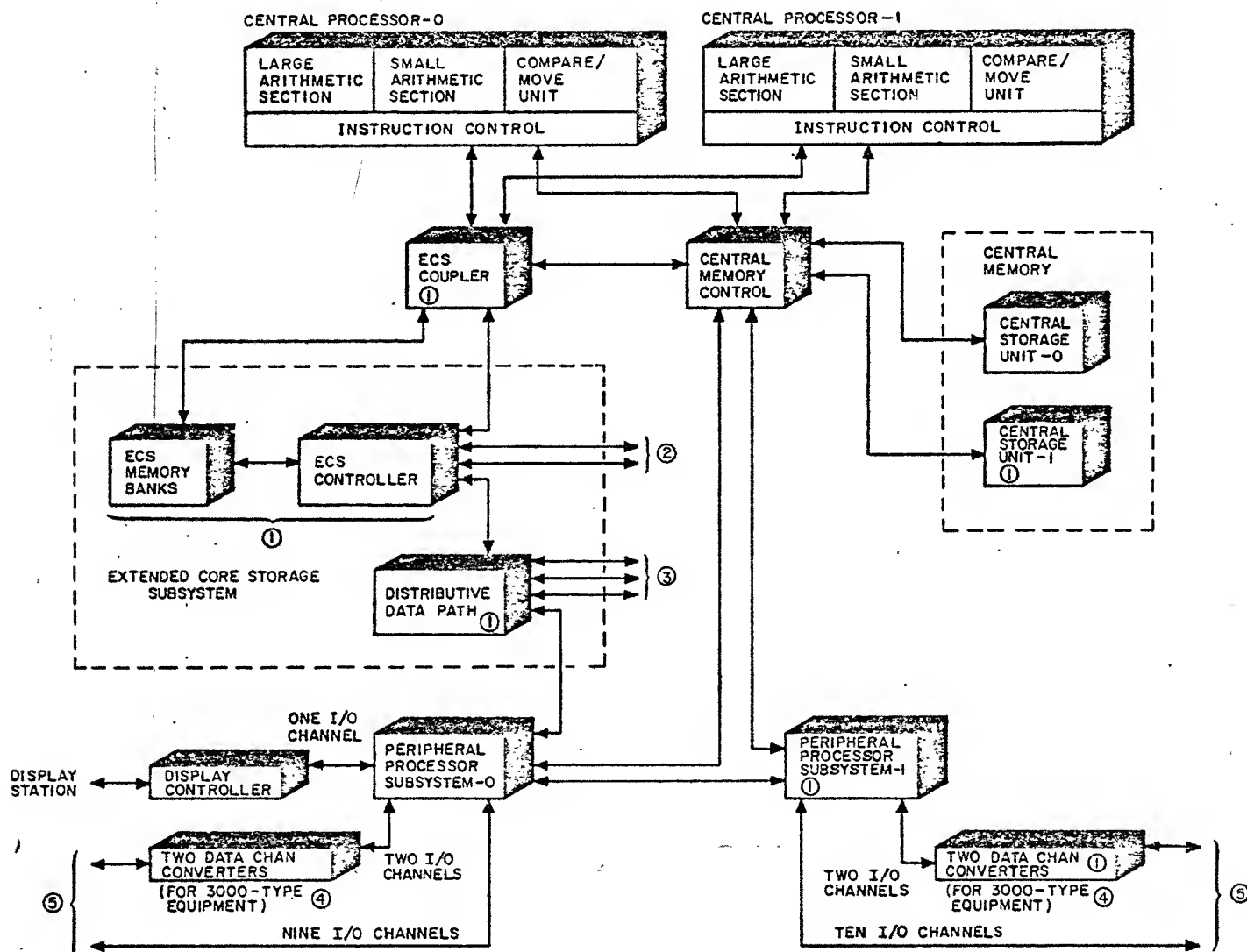
## EXTENDED CORE STORAGE

- Organization, addressing, phasing
- Access, degradation, flag register
- Controller coupler transfer rates
- Distributed datapath operation

## MODEL 174 SYSTEM

The model 174 basic computer system (figure 1-12) is functionally similar to model 173, except that the

system provides faster operation. Model 174 differs basically from model 173 by having a second CP. The ECS, CM, PPS, and I/O options are the same as for model 173.



### NOTES:

- ① OPTIONAL EQUIPMENT.
- ② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS.
- ③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS.
- ④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN PPS-0 AND PPS-1.
- ⑤ PERIPHERAL EQUIPMENT.

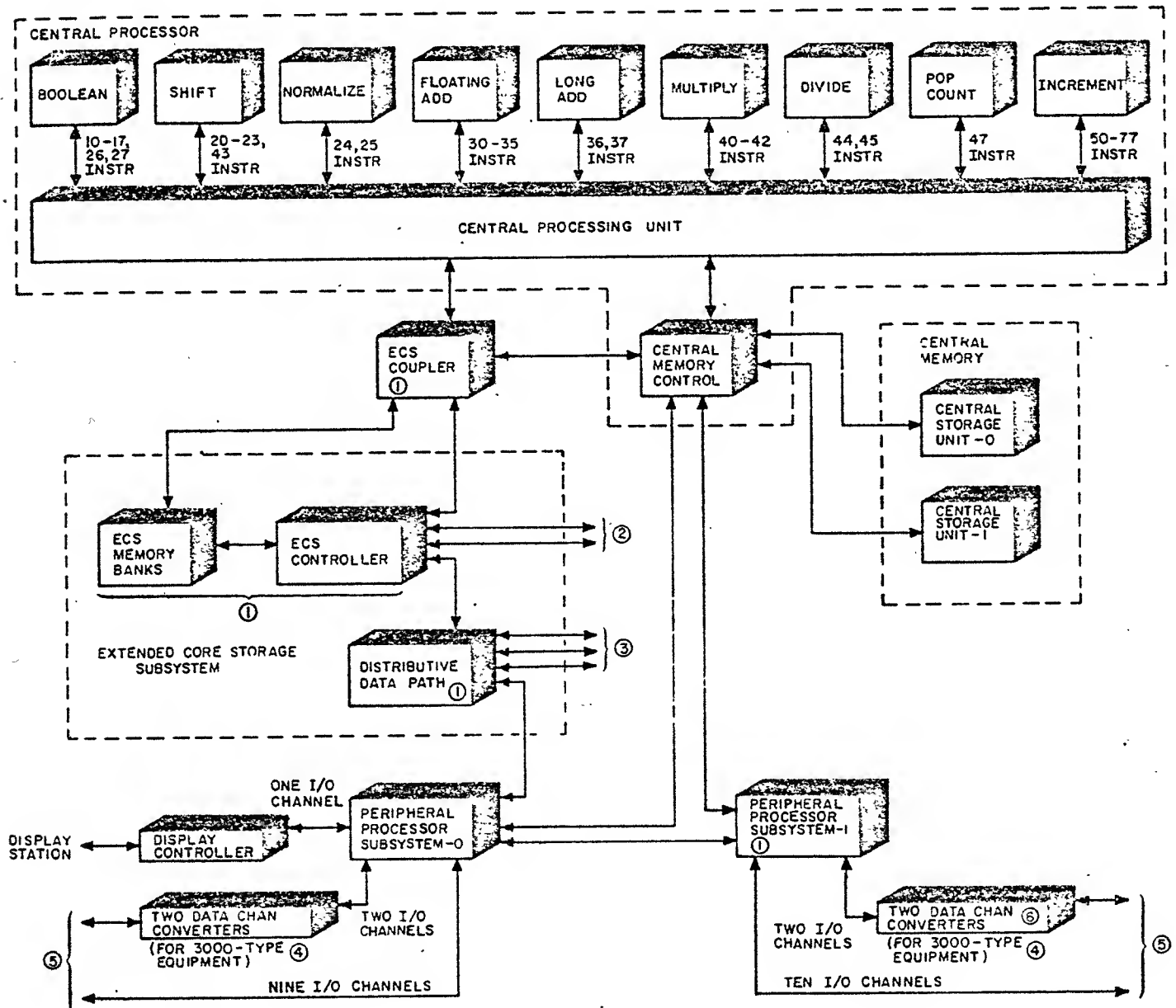
Figure 1-12. Model 174 Computer System

3AR130

## MODEL 175 SYSTEM

The model 175 basic computer system (figure 1-13) is functionally similar to model 173 and its options except in the CP. In place of the serial CP, the model 175 CP contains nine functional units, a cen-

tral processing unit (CPU), and the CMC. The nine functional units operate in parallel as independent specialized arithmetic units, providing maximum overlap of instruction retrieval and execution. The basic model 175 has two CSUs that provide 16 independent banks of memory.



### NOTES:

- ① OPTIONAL EQUIPMENT.
- ② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS.
- ③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS.
- ④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN THE PPS.
- ⑤ PERIPHERAL EQUIPMENT.
- ⑥ OPTIONAL EQUIPMENT FOR MODELS 175A AND 175B. NOT AVAILABLE FOR MODEL 175C

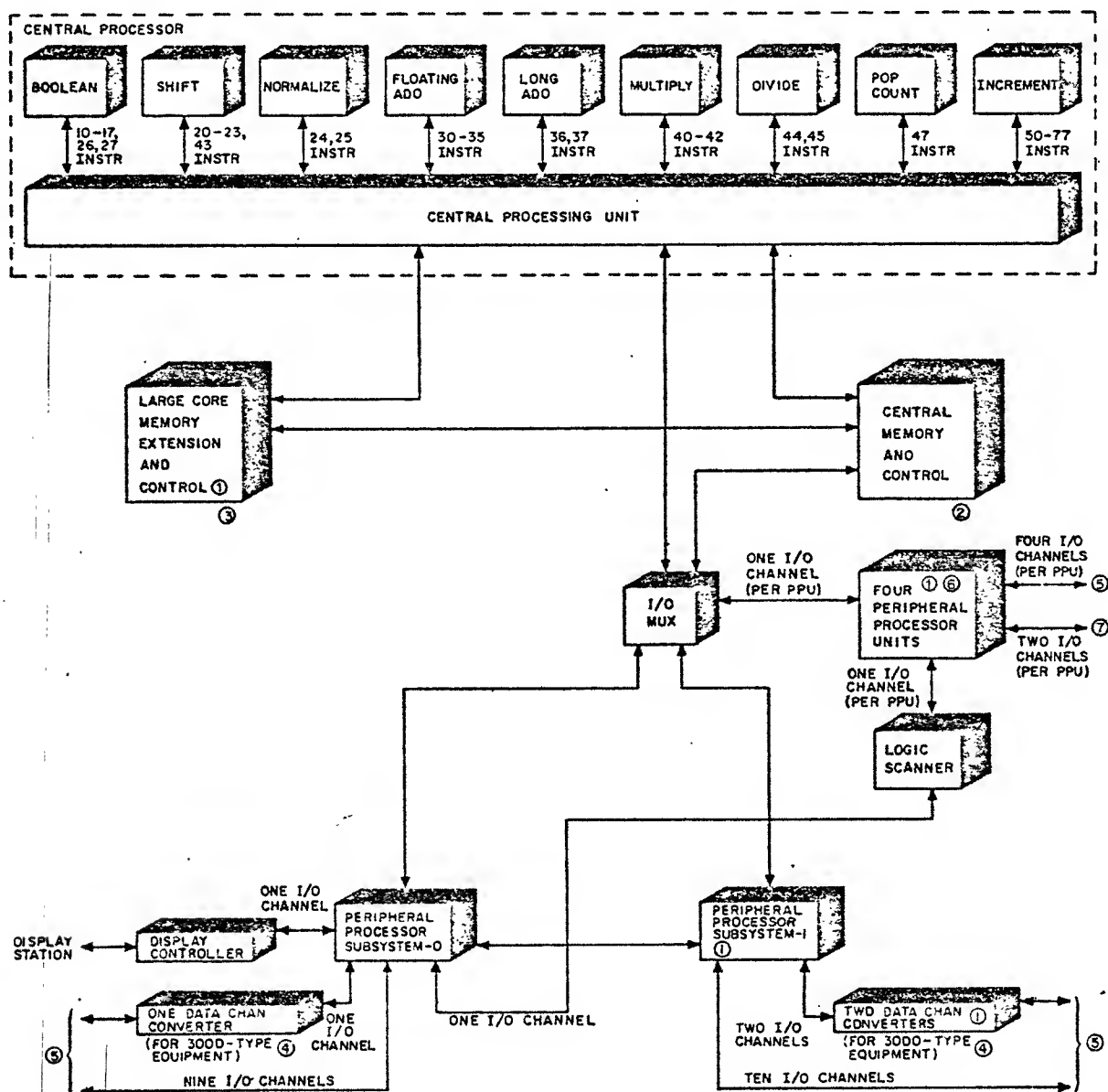
3AR14E

Figure 1-13. Model 175 Computer System

## MODEL 176 SYSTEM

The model 176 basic computer system (figure 1-14) is functionally similar to model 175 in the areas of the CP and PPS. Model 176 differs basically from model 175 in the use of an LCME option in the basic system instead of having an ECS option. The CM is

still optionally expandable but does not have separate CSUs as in other models. The CM and LCME each contain their own control functions. Other major differences include the option of adding from 4 to a total of 13 PPUs, an I/O multiplexer, and a logic scanner to permit PPS communication with the PPUs.



### NOTES:

- ① OPTIONAL EQUIPMENT.
- ② BASIC CM CONTAINS 16 INDEPENDENT BANKS OF 8,192 60-BIT WORDS EACH AND IS EXPANDABLE TO 12,288 TO 16,384 WORDS.
- ③ BASIC LCME CONTAINS 2 INDEPENDENT BANKS OF 262,144 60-BIT WORDS AND IS EXPANDABLE TO 4 OR 8 INDEPENDENT BANKS OF 262,144 WORDS PER BANK.
- ④ ADDITIONAL DATA CHANNEL CONVERTERS MAY BE ADDED EXTERNALLY TO PPS-0 AND PPS-1.
- ⑤ PERIPHERAL EQUIPMENT.
- ⑥ ADDITIONAL PERIPHERAL PROCESSOR UNITS ARE OPTIONAL.
- ⑦ ONE CHANNEL FOR INTER-PPU COMMUNICATIONS AND ONE CHANNEL NOT USED.

3AR10C

Figure 1-14. Model 176 Computer System

## FUNCTIONAL CHARACTERISTICS

Tables 1-2 through 1-6 summarize the functional characteristics of the CP, CM, PPS, and data address and checking for each system.

TABLE 1-2. CENTRAL PROCESSOR FUNCTIONAL CHARACTERISTICS

Functional Characteristics	Models					
	171	172	173	174	175	176
60-bit internal word	x	x	x	x	x	x
Computation in fixed- and floating-point arithmetic	x	x	x	x	x	x
Eight 60-bit operand X registers	x	x	x	x	x	x
Eight 18-bit address A registers	x	x	x	x	x	x
Eight 13-bit index B registers	x	x	x	x	x	x
Character manipulation by compare/move instructions	*	x	x	x	-	-
Synchronous internal logic with a 50-nanosecond clock period	x	x	x	x	-	-
Synchronous internal logic with a 27.5-nanosecond clock period	-	-	-	-	-	x
Large and small arithmetic sections	x	x	x	x	-	-
Synchronous internal logic with a 25-nanosecond clock period	-	-	-	-	x	-
12-word instruction word stack	-	-	-	-	x	x
Nine functional units	-	-	-	-	x	x
x Standard - Not available * Optional						

TABLE 1-3. CENTRAL MEMORY FUNCTIONAL CHARACTERISTICS

Functional Characteristics	Model					
	171	172	173	174	175	176
400-nanosecond cycle time for all models except model 175C series 300 which has a 300-nanosecond cycle time	x	x	x	x	x	-
165-nanosecond cycle time for write, 82.5-nanosecond cycle time for read	-	-	-	-	-	x
Maximum transfer rate of one word each 50 nanoseconds	x	x	x	x	x	-
Maximum transfer rate of one word each 27.5 nanoseconds	-	-	-	-	-	x
Semiconductor memory of 65,536 words (60-bit words plus eight error detection/correction bits per word); expandable to 98,304; 131,072; 196,608; and 262,144 words	x	x	x	x	-	-
Semiconductor memory of 131,072 words (60-bit words plus eight error detection/correction bits per word); expandable to 196,608 and 262,144 words	-	-	-	-	x	x
Organized into eight independent banks per CSU	x	x	x	x	-	-
Organized into 16 independent banks	-	-	-	-	x	x
x Standard - Not available						

TABLE 1-4. PERIPHERAL PROCESSOR SUBSYSTEM FUNCTIONAL CHARACTERISTICS

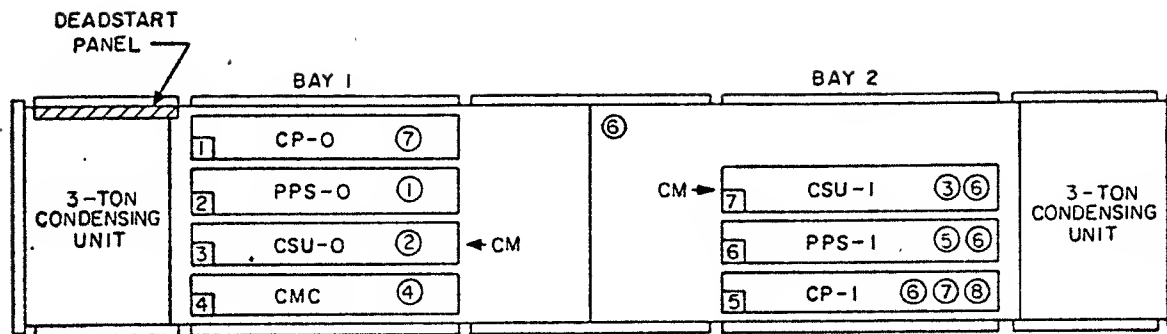
Functional Characteristics	Model					
	171	172	173	174	175	176
12-bit internal word	x	x	x	x	x	x
Binary computation in fixed-point arithmetic	x	x	x	x	x	x
Selectable operating speeds of 1X or 2X (1X equals major cycle of 1000 nanoseconds and minor cycle of 100 nanoseconds; 2X equals major cycle of 500 nanoseconds and minor cycle of 50 nanoseconds)	x	x	x	x	x	x
10 PPs time-share access to CM	x	x	x	x	x	x
Each PP has an internal semiconductor memory of 4096 words (12-bit words plus one parity bit per word, odd parity)	x	x	x	x	x	x
12 I/O channels, each accessible by any of the PPs	x	x	x	x	x	x
Status and control register	x	x	x	x	x	x
Real-time clock	x	x	x	x	x	x
Each I/O channel carries 12-bit words plus one parity bit per word (odd parity)	x	x	x	x	x	x
Expandable from 10 to 20 PPs in increments of 4, 3, and 3 and from 12 to 24 I/O channels	*	*	*	*	*	*
x Standard - Not available * Optional						

TABLE 1-5. PERIPHERAL PROCESSOR UNIT FUNCTIONAL CHARACTERISTICS

Functional Characteristics	Model					
	171	172	173	174	175	176
12-bit internal word	-	-	-	-	-	x
Binary computation in fixed-point arithmetic	-	-	-	-	-	x
27.5-nanosecond clock synchronous with CP	-	-	-	-	-	x
Each PPU has an internal coincident current memory of 4096 words (12-bit words plus one parity bit per word, odd parity)	-	-	-	-	-	x
Eight bidirectional I/O channels dedicated to each PPU	-	-	-	-	-	x
x Standard - Not available						

TABLE 1-6. DATA AND ADDRESS CHECKING FUNCTIONAL CHARACTERISTICS

Functional Characteristics	Model					
	171	172	173	174	175	176
Parity check data between CP-0 and CMC	x	x	x	x	-	-
Parity check data between CP-1 and CMC	x	x	-	x	-	-
Parity check data between PPS-0 and CMC	x	x	x	x	x	-
Parity check data between PPS-1 and CMC	x	x	x	x	x	-
Parity check data between ECS and CMC (only if a parity-enhanced controller is installed)	x	x	x	x	x	-
Single-error correction double-error detection (SECEDED) between CM and CMC	x	x	x	x	x	-
SECEDED between CM and control	-	-	-	-	-	x
Parity check address from CP-0 to CMC	x	x	x	x	-	-
Parity check address from CP-1 to CMC	x	x	-	x	-	-
Parity check address from PPS-0 to CMC	x	x	x	x	x	-
Parity check address from PPS-1 to CMC	x	x	x	x	x	-
Parity check address from CMC to CM	x	x	x	x	x	-
Parity check data between CM and control (non-SECEDED mode only)	x	x	x	x	x	x
SECEDED between LCME and LCME control	-	-	-	-	-	x
Parity check data between LCME and LCME control (non-SECEDED mode only)	-	-	-	-	-	x
Parity check on PPS memory data	x	x	x	x	x	x
Parity check on PPU memory data	-	-	-	-	-	x
x Standard - Not available						

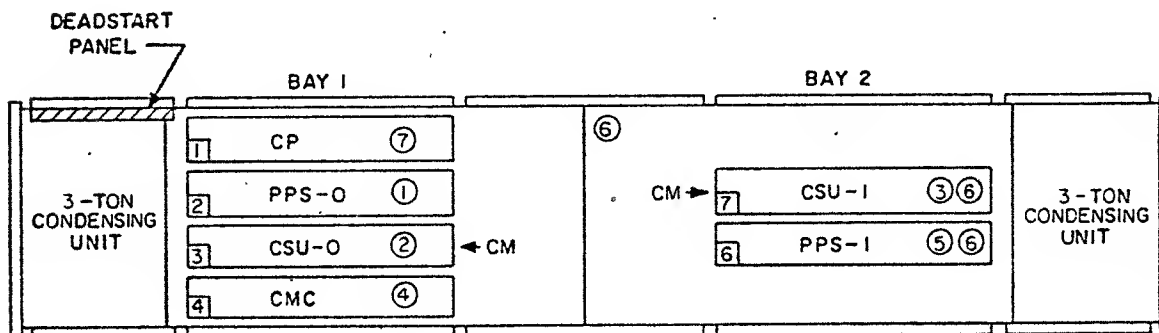


NOTES:

- ① CHASSIS 2 ALSO CONTAINS TWO DATA CHANNEL CONVERTERS AND A DISPLAY STATION CONTROLLER.
- ② CSU-0 IS EXPANDABLE FROM 65,536 TO 98,304 TO 131,072 WORDS.
- ③ CSU-1 EXPANDS MEMORY TO 196,608 TO 262,144 WORDS.
- ④ WHEN ECS IS INSTALLED, CHASSIS 4 ALSO CONTAINS THE ECS COUPLER.
- ⑤ CHASSIS 6 MAY CONTAIN TWO OPTIONAL DATA CHANNEL CONVERTERS.
- ⑥ BAY 2 AND CHASSIS 5, 6, AND 7 ARE OPTIONAL.
- ⑦ CHASSIS 1 AND 5 EACH CONTAIN A COMPARE/MOVE UNIT.
- ⑧ IF CHASSIS 5 IS INSTALLED AND CHASSIS 7 IS NOT INSTALLED, CHASSIS 6 MOUNTS IN THE CHASSIS 7 LOCATION.

6AA3D

Figure 1-3. Model 172 Maximum Chassis Configuration (Top Cutaway View)



NOTES:

- ① CHASSIS 2 ALSO CONTAINS TWO DATA CHANNEL CONVERTERS AND A DISPLAY STATION CONTROLLER.
- ② CSU-0 IS EXPANDABLE FROM 65,536 TO 98,304 TO 131,072 WORDS.
- ③ CSU-1 EXPANDS MEMORY TO 196,608 TO 262,144 WORDS.
- ④ WHEN ECS IS INSTALLED, CHASSIS 4 ALSO CONTAINS THE ECS COUPLER.
- ⑤ CHASSIS 6 MAY CONTAIN TWO OPTIONAL DATA CHANNEL CONVERTERS.
- ⑥ BAY 2 AND CHASSIS 6 AND 7 ARE OPTIONAL.
- ⑦ CHASSIS 1 ALSO CONTAINS A COMPARE/MOVE UNIT.

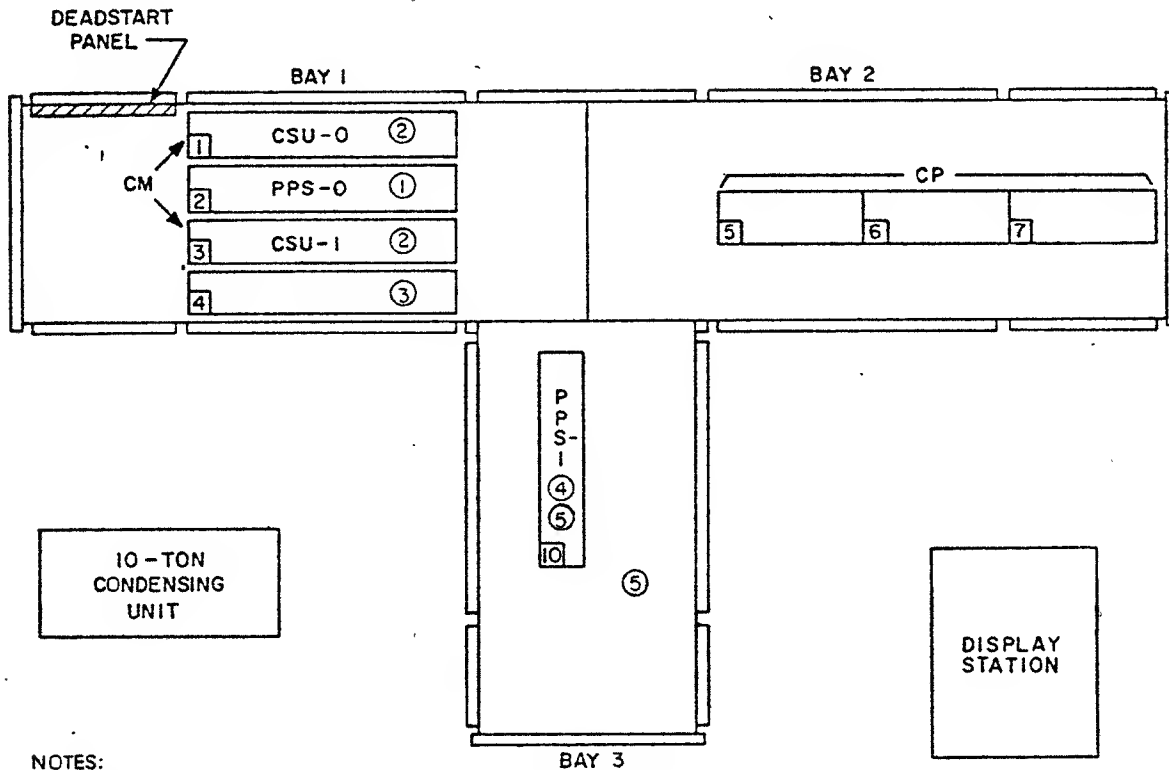
6AA4D

Figure 1-4. Model 173 Maximum Chassis Configuration (Top Cutaway View)

## MODEL 175 CONFIGURATION

The models 175 A and B basic configuration (figure 1-6) includes a display station, a stand-alone condensing unit, and mainframe bays 1 and 2, which contain four chassis in bay 1 and three chassis in bay 2.

The maximum configuration includes one additional chassis in bay 3. Installation of the optional ECS requires the addition of a stand-alone cabinet for a controller and from one to four cabinets for the ECS, depending upon the options.



### NOTES:

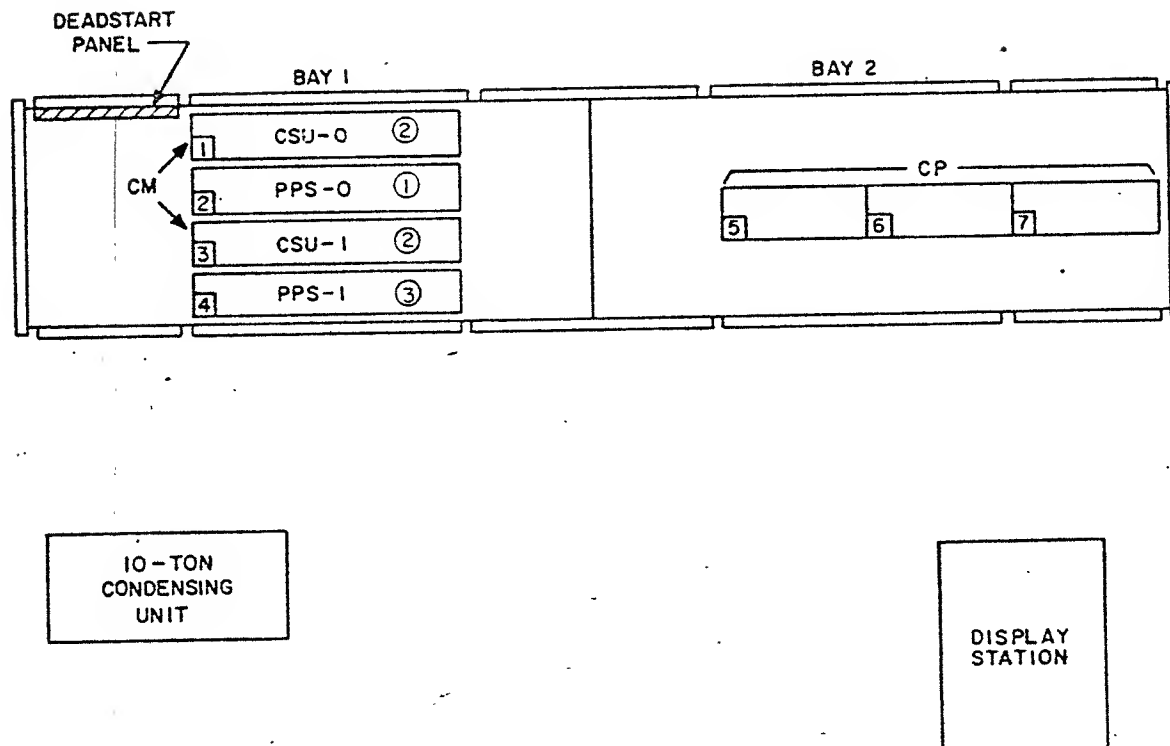
- ① CHASSIS 2 ALSO CONTAINS TWO DATA CHANNEL CONVERTERS AND A DISPLAY STATION CONTROLLER.
- ② CSU CHASSIS ARE EXPANDABLE FROM 65,536 TO 98,304 TO 131,072 TO 196,608 TO 262,144 WORDS.
- ③ WHEN ECS IS INSTALLED, CHASSIS 4 CONTAINS THE ECS COUPLER.
- ④ CHASSIS 10 MAY CONTAIN TWO OPTIONAL DATA CHANNEL CONVERTERS.
- ⑤ BAY 3 AND CHASSIS 10 ARE OPTIONAL.

6A46E

Figure 1-6. Models 175A and B Maximum Chassis Configuration (Top Cutaway View)

The model 175C basic configuration (figure 1-7) includes a display station, a stand-alone condensing unit, and mainframe bays 1 and 2, which contain four chassis in bay 1 and three chassis in bay 2. The maximum configuration has the same bays

and chassis except for options added to the chassis. Installation of the optional ECS requires the addition of a stand-alone cabinet for a controller and from one to four cabinets for the ECS, depending upon the options.



NOTES:

- ① CHASSIS 2 ALSO CONTAINS TWO DATA CHANNEL CONVERTERS AND A DISPLAY STATION CONTROLLER.
- ② CSU CHASSIS ARE EXPANDABLE FROM 65,536 TO 98,304 TO 131,072 TO 196,608 TO 262,144 WORDS.
- ③ WHEN ECS IS INSTALLED, CHASSIS 4 CONTAINS THE ECS COUPLER.

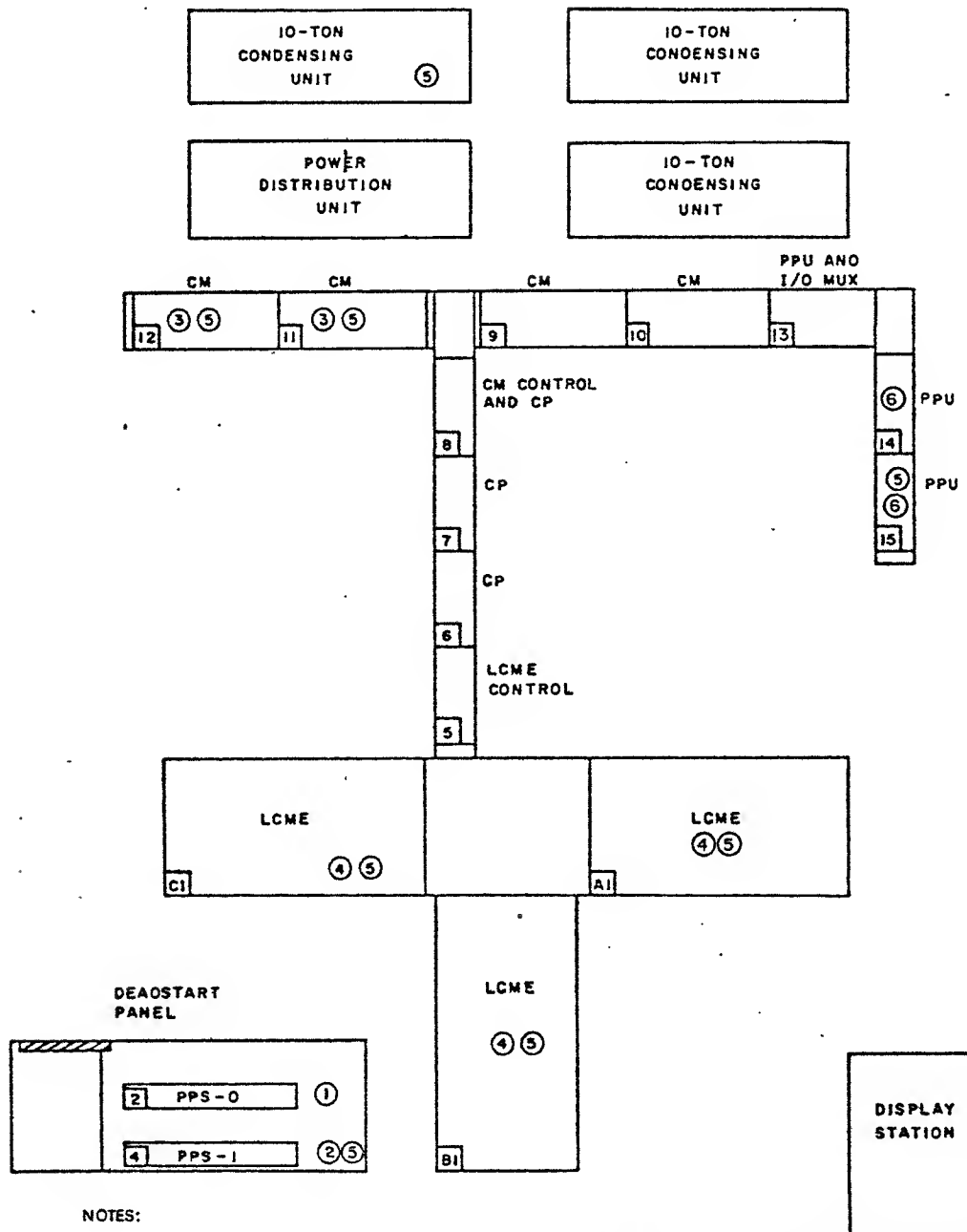
6AA24A

Figure 1-7. Model 175C Maximum Chassis Configuration (Top Cutaway View)

## MODEL 176 CONFIGURATION

The model 176 basic configuration (figure 1-8) includes a display station, two condensing units, a stand-alone cabinet with one chassis, and eight

mainframe chassis. The maximum configuration includes one additional condensing unit, one additional chassis in the stand-alone cabinet, and six additional mainframe chassis.



### NOTES:

- ① CHASSIS 2 ALSO CONTAINS ONE DATA CHANNEL CONVERTER AND A DISPLAY STATION CONTROLLER.
- ② CHASSIS 4 MAY CONTAIN TWO OPTIONAL DATA CHANNEL CONVERTERS.
- ③ CM IS EXPANDABLE FROM 131,072 TO 196,608 TO 262,144 WORDS.
- ④ LCME IS OPTIONAL BEGINNING AT 524,288 WORDS AND EXPANDABLE TO 1,048,576 OR 2,097,152 WORDS.
- ⑤ CHASSIS 4, 11, 12, 15, A1, B1, C1, AND ONE 10-TON CONDENSING UNIT ARE OPTIONAL.
- ⑥ PPU'S ARE OPTIONAL.

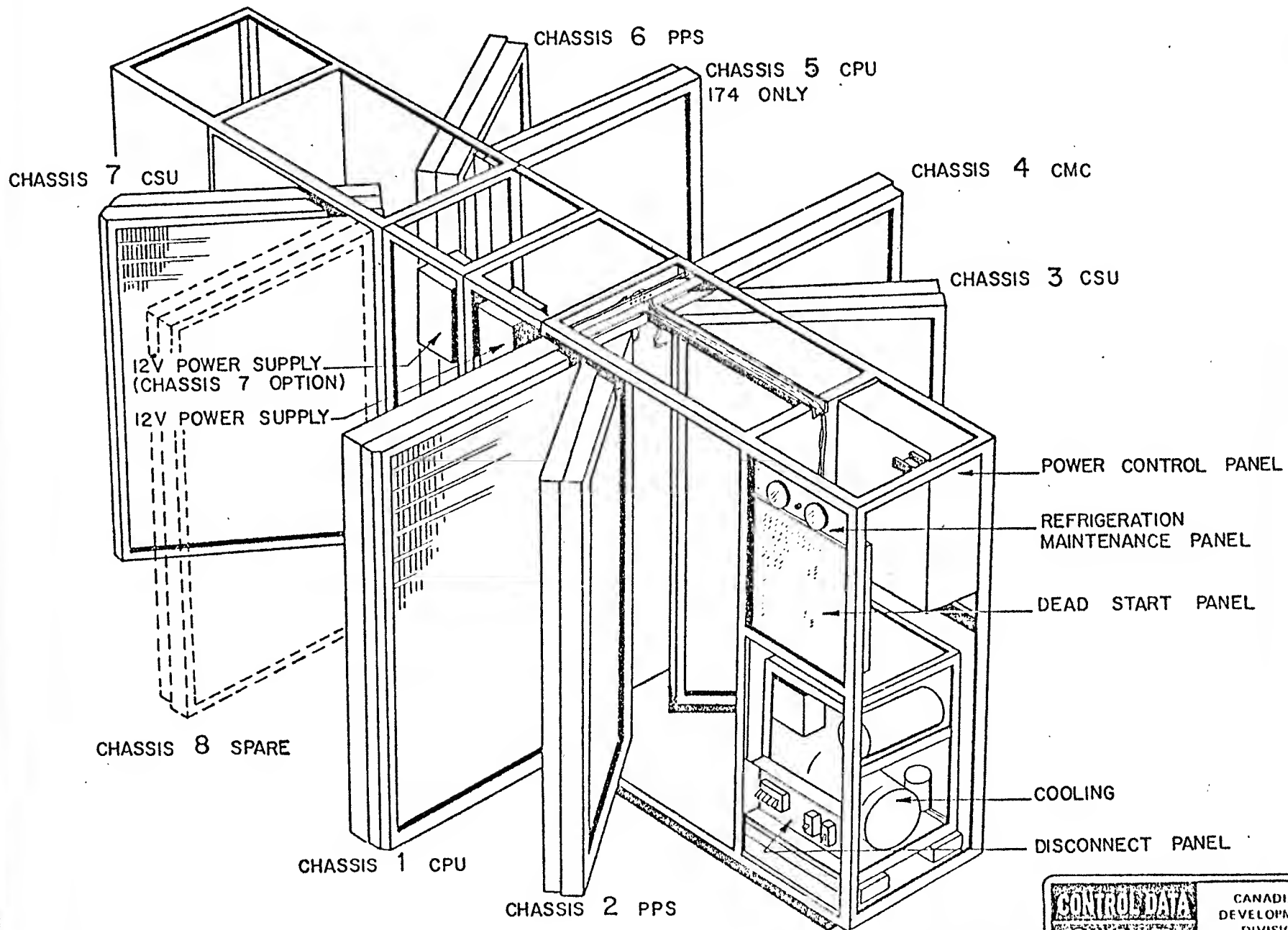
6AA218

Figure 1-8. Model 176 Maximum Chassis Configuration (Top Cutaway View)

# CDC CYBER 173 / 174

CYBER 173 MAX. CONFIGURATION  
USES CHASSIS 1,2,3,4,6,7

CYBER 174 MAX. CONFIGURATION  
USES CHASSIS 1,2,3,4,5,6,7



APRIL 75

**CONTROL DATA**  
CANADA

CANADIAN  
DEVELOPMENT  
DIVISION

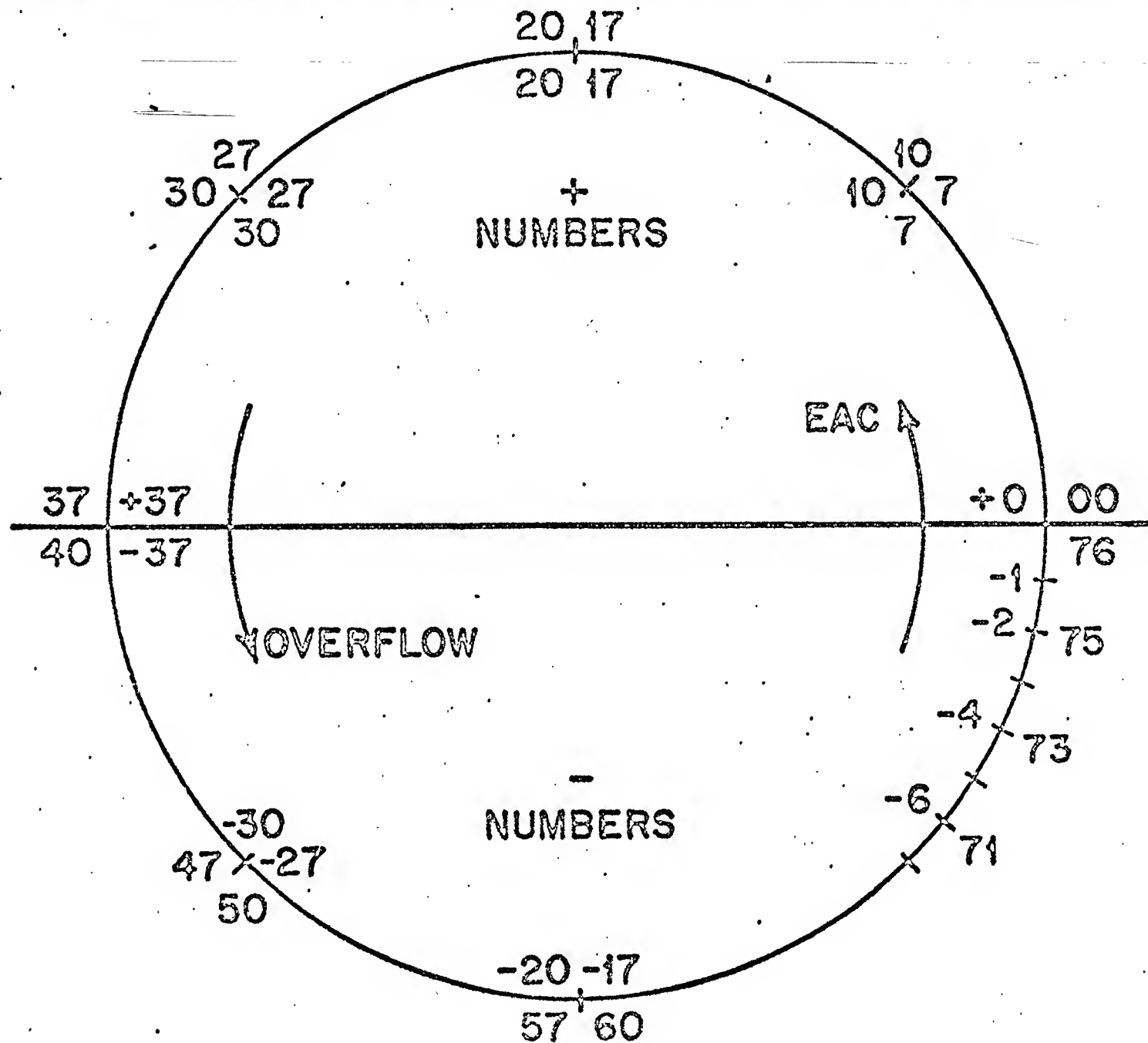
BPA	Breakpoint address	MOS	Metal oxide semiconductor
CEJ	Central exchange jump	NEA	Normal exit address
CIW	Current instruction word	P	Program address
CM	Central memory	PE	Parity error
CMC	Central memory control	PP	Peripheral processor
CP	Central processor	PPM	Peripheral processor memory
CPU	Central processing unit	PPS	Peripheral processor subsystem
CSU	Central storage unit	PPU	Peripheral processor unit
ECS	Extended core storage	PSD	Program status designator
EEA	Exit error address	RAC	Reference address for CM
EM	Exit mode	RAE	Reference address for ECS
FLC	Field length for CM	RAL	Reference address for LCME
FLE	Field length for ECS	RAS	Reference address for CM
FLL	Field length for LCME	RNI	Read next instruction
FLS	Field length of program for CM	RVM	Reference voltage margin
IAS	Instruction address stack	SAS	Storage address stack
IFA	Instruction fetch address	SECDED	Single-error correction double-error detection
I/O	Input/output	SRO	Storage read out
IWS	Instruction word stack	SWS	Storage word stack
LCME	Large core memory extension		
MA	Monitor address		
MEJ	Monitor exchange jump		
MF	Monitor flag		

## NOTE

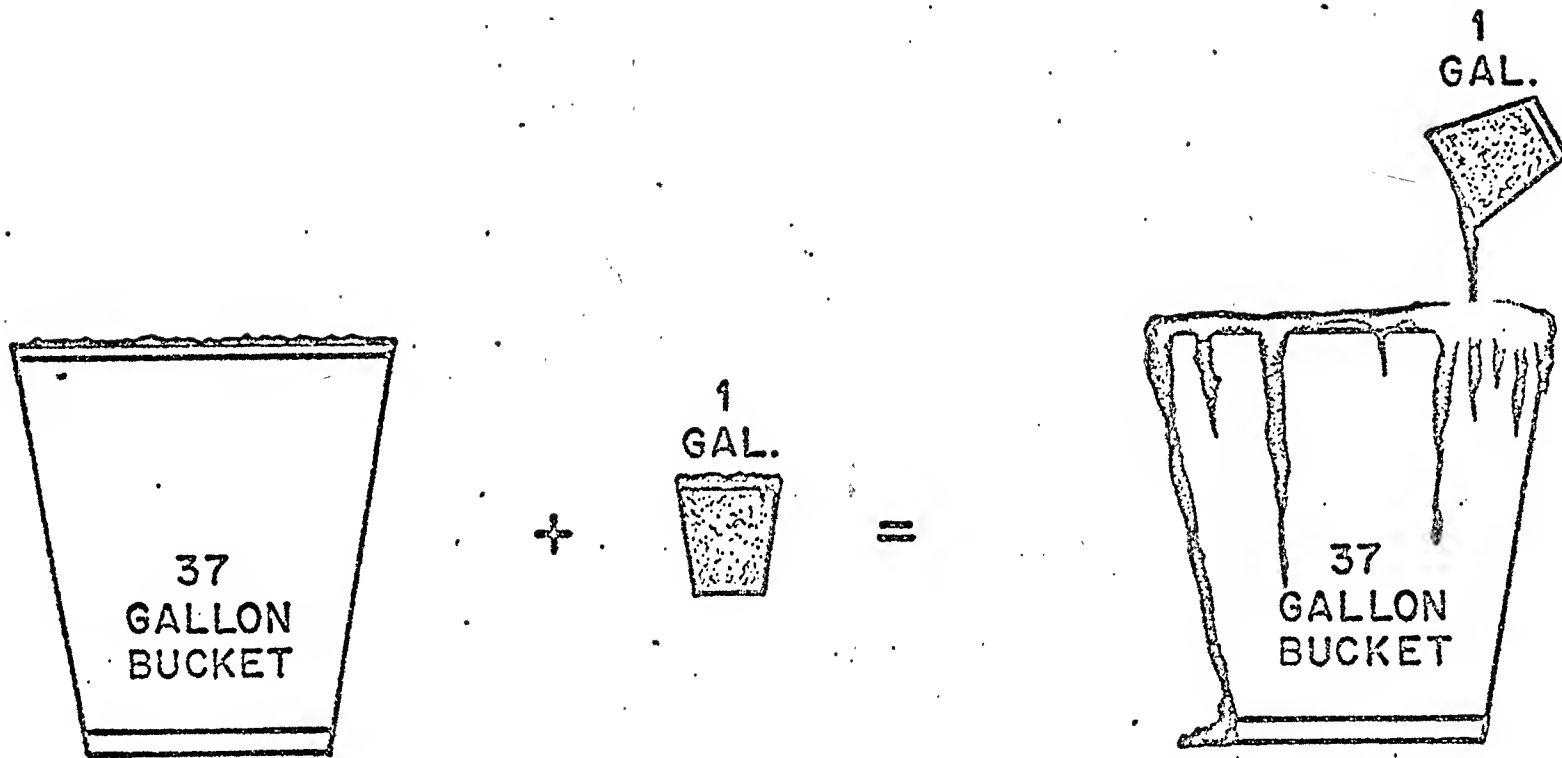
Instruction designators are defined in tables 4-1 and 4-7.

3

# MODULUS WHEEL - CLOSED (ONE'S COMPLEMENT) DEVICE



# OVERFLOW



$$\begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}_2 + \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}_2 = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}_2$$

The diagram shows a binary addition problem. The first number is 011111<sub>2</sub> and the second number is 000001<sub>2</sub>. The result is 100000<sub>2</sub>. A carry bit of 1 is shown above the first digit of the result.

OVERFLOW.

ADDITIVE

$$\begin{array}{r} (+) \quad + \\ \quad + \\ \hline - \end{array} \quad \begin{array}{r} (+) \quad - \\ \quad - \\ \hline + \end{array}$$

SUBTRACTIVE

$$\begin{array}{r} (-) \quad + \\ \quad - \\ \hline - \end{array} \quad \begin{array}{r} (-) \quad - \\ \quad + \\ \hline + \end{array}$$

# BINARY ADDITION

RULES:

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 0 \end{array}$$

AND A CARRY

SATISFY

PASS

PASS

GENERATE

EXAMPLE:

0	0	1	1	1	0
---	---	---	---	---	---

CARRIES

1	0	0	0	1	1
---	---	---	---	---	---

ADDEND

0	1	0	1	0	1
---	---	---	---	---	---

AUGEND

---

1	1	1	0	0	0
---	---	---	---	---	---

SUM

## 6000 ADDER DEFINITIONS

Stage - two bits from two operands to be arithmetically combined, of equal binary weight.

Generate - a carry or borrow will be Generated by this stage combination to the next stage regardless of it's input

Pass - a carry/borrow will be passed to the next stage only if a generate-input occurred at the pass stage

Satisfy - a stage combination which does not allow a carry/borrow to be input to the next stage regardless of input

Generate - a carry {Add} or borrow {Subtract} outputted from a particular stage or group

Generate-Input - a carry/borrow input, ed to a particular stage or group

	ADDER		SUBTRACTOR	
	True	Comp	True	Comp
Generate	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>
	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>
Pass	<u>01</u>	<u>10</u>	<u>10</u>	<u>01</u>
	<u>10</u>	<u>01</u>	<u>10</u>	<u>01</u>
Satisfy	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>
	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>

## Generate Logic

Operand - A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Operand - B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stage	2 <sup>17</sup>	2 <sup>16</sup>	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
Group	5			4			3			2		1		0				
Section	1						0											

$$\begin{array}{r}
 \text{MULTIPLICAND (Md)} \\
 \times \text{MULTIPLIER (Mr)} \\
 \hline
 \text{PRODUCT (Pr)}
 \end{array}$$

### MULTIPLY ALGORITHM

1. Convert Mr and Md to positive operands. (Note original signs.)
2. Check the LSD of the Mr if it is;  
 "1" = Add the Md to the Pr  
 "0" = Add zero's to the Pr (do Nothing)
3. Left shift the Md (shift a zero into the LSD).
4. Right shift the Mr (End Off).
5. Repeat steps 2 through 4 until the Mr is reduced to nothing. At this point check the signs of the original Md and Mr if;  
 = Pr is the correct result  
 ≠ Complement Pr to obtain the correct result.

### EXAMPLES

$$5_{10} \times 5_{10} = 25_{10}$$

	Md	Mr	Pr
Start	000101	000101	000 000
Step 1	000101	000101	000000
Step 2		000101	000 000 + 000101 000101
Step 3-4	001010	00010	-
Step 2	-	00010	000 101
Step 3-4	010100	0001	-
Step 2	-	0001	000101 + 010100 011001
Step 3-4	101000		
Step 5			011001

$$-7_{10} \times 6_{10} = -42_{10}$$

	Md	Mr	Pr
Start	0110	1000	0000
Step 1	110	111	000
Step 2	-	111	000 +110 110
Step 3-4	1100	11	-
Step 2		11	110 +1100 10010
Step 3-4	11000	1	-
Step 2	-	1	10010 +11000 101010
Step 3-4	110000	( )	-
Step 5	0101010	→	1010101

# DIVIDE ALGORITHM {WHOLE NUMBERS}

1. Write the Divisor {Dv} and the Remainder {Rn} in true form. Set the sign digit in the quotient to "0" {Q = 0\_\_\_. ----}
2. Shift the Dv left until it's MSD has the same binary weight as the MSD OF the Remainder Ro {Note the number of bits that the divisor is shifted for Quotient correction in Step 5}
3. Subtract the Divisor from the Remainder if:
  - a. EAB occurs, set the next Quotient digit to "0" and use the remainder {minuend} for Step 4.
  - b. EAB occurred, set the next Quotient digit to "1" and use the difference as the "remainder" for Step 4.
4. Left shift the Remainder one bit position {Shift in a zero}
5. Repeat item 3 and 4 until either the desired accuracy has been achieved, or the remainder in Step 3a has been reduced to zero. At this point:
  - a. Shift the Quotient left the number of places which the divisor was shifted in Step 2 {This correctly positions the radix point in the result}
  - b. Check the sign digits of the original remainder and divisor, if they were:
    - = The Quotient from Step 5-a is the correct result
    - ≠ Complement the Quotient from Step 5-a to obtain the correct result

EXAMPLE? DIVIDE  $\frac{25_{10}}{5_{10}} = \frac{31_8}{5_8} = 5$

Step	Divisor	Remainder	Quotient
1	0 1 0 1	0 1 1 0 0 1	0 -.-.-
2	0 1 0 1 0 0	{Same}	{Same} Dv Shift 2
3b	{Same}	$\begin{array}{r} 0\ 1\ 1\ 0\ 0\ 1 \\ -\ 0\ 1\ 0\ 1\ 0\ 0 \\ \hline 0\ 0\ 0\ 1\ 0\ 1 \end{array}$	0 1. - - -
4	{Same}	, 0 0 0 1 0 1 0	{Same}
3a	{Same}	$\begin{array}{r} 0\ 0\ 0\ 1\ 0\ 1\ 0 \\ -\ 0\ 0\ 1\ 0\ 1\ 0\ 0 \\ \hline \text{EAB} \end{array}$	0 1.0 - -

# DIVIDE ALGORYTHM CONT

Step	Divisor	Remainder	Quotient
4	{Same}	0 0 0 1 0 1 0 <u>0</u>	{Same}
3b	{Same}	<div> 0 0 0 1 0 1 0 0  -       0 1 0 1 0 0  -----  0 0 0 0 0 0 0 0 </div>	0 1.0 1 -
5-a,b	{Same}	- - -	0 1 0 1. {LS-2}

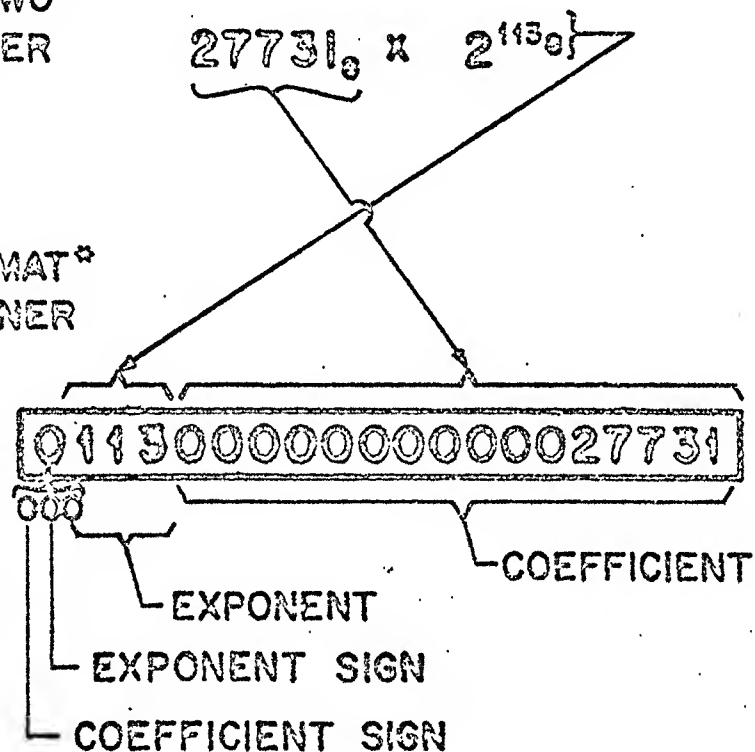
# 6000 SERIES FLOATING POINT FORMAT <sup>33</sup>

FOLLOWING IS AN EXAMPLE OF HOW A NUMBER WITH A  
LARGE MAGNITUDE WOULD BE EXPRESSED IN A 6000  
SERIES COMPUTER WORD. 1

**277340000000000000000000000000000,**

WHICH CAN BE EXPRESSED  
IN OCTAL POWERS OF TWO  
IN THE FOLLOWING MANNER

AND IN THE 6000 FORMAT\*  
IN THE FOLLOWING MANNER



**\* UNBIASED**

POSITIVE NUMBER, POSITIVE EXPONENT

$$2 \times 2^3 = 200300000000000000002$$

NEGATIVE NUMBER, POSITIVE EXPONENT

$$-(2 \times 2^3) = 577477777777777777775$$

POSITIVE NUMBER, NEGATIVE EXPONENT

$$2 \times 2^{-3} = 177400000000000000002$$

NEGATIVE NUMBER, NEGATIVE EXPONENT

$$-(2 \times 2^{-3}) = 600377777777777777775$$

## NUMBERS IN THE CDC 6000

48 - 58	0 - 47
---------	--------

48-bit  
Coefficient

Biased Exponent	0000	0001	0002	...	1776	1777	2000	2001	...	3776	3777
Exponent	----	-1776	-1775	...	0001	----	0000	0001	...	1776	----

Now suppose we have a positive real finite, definite number with coefficient  $F$  and Biased Exponent  $C$ . Let  $E$  be the exponent associated with  $C$ . Then the value of our number is  $F$  times  $2^E$ .  $F$  is a 48-bit integer, its value is between 0 and  $2^{48} - 1$  inclusive.

20000000000000000000  
17204000000000000000  
77774000000000000000  
60503000000000000000  
17165252525252525252  
1716314631463146314

```

1 times 20 = 1.0
400000000000000000 {octal} times 2-47 = 1.0
400000000000000000 {octal} times 2-48 = 0.5
the negative of the number above = -0.5
approximately 1/3
approximately 1/5

```



# FUNDAMENTAL RULES OF BOOLEAN ALGEBRA

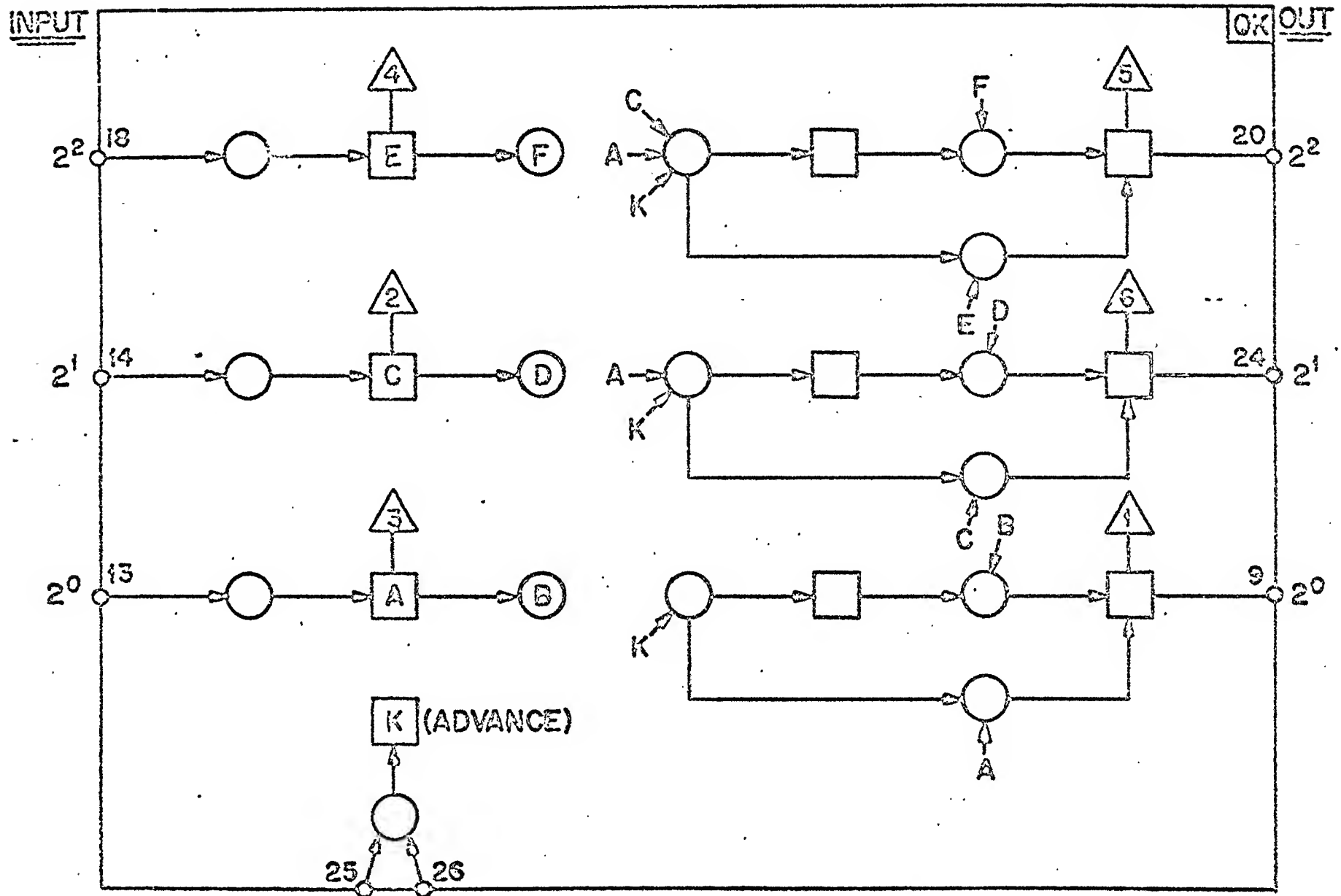
$\cdot$  = AND

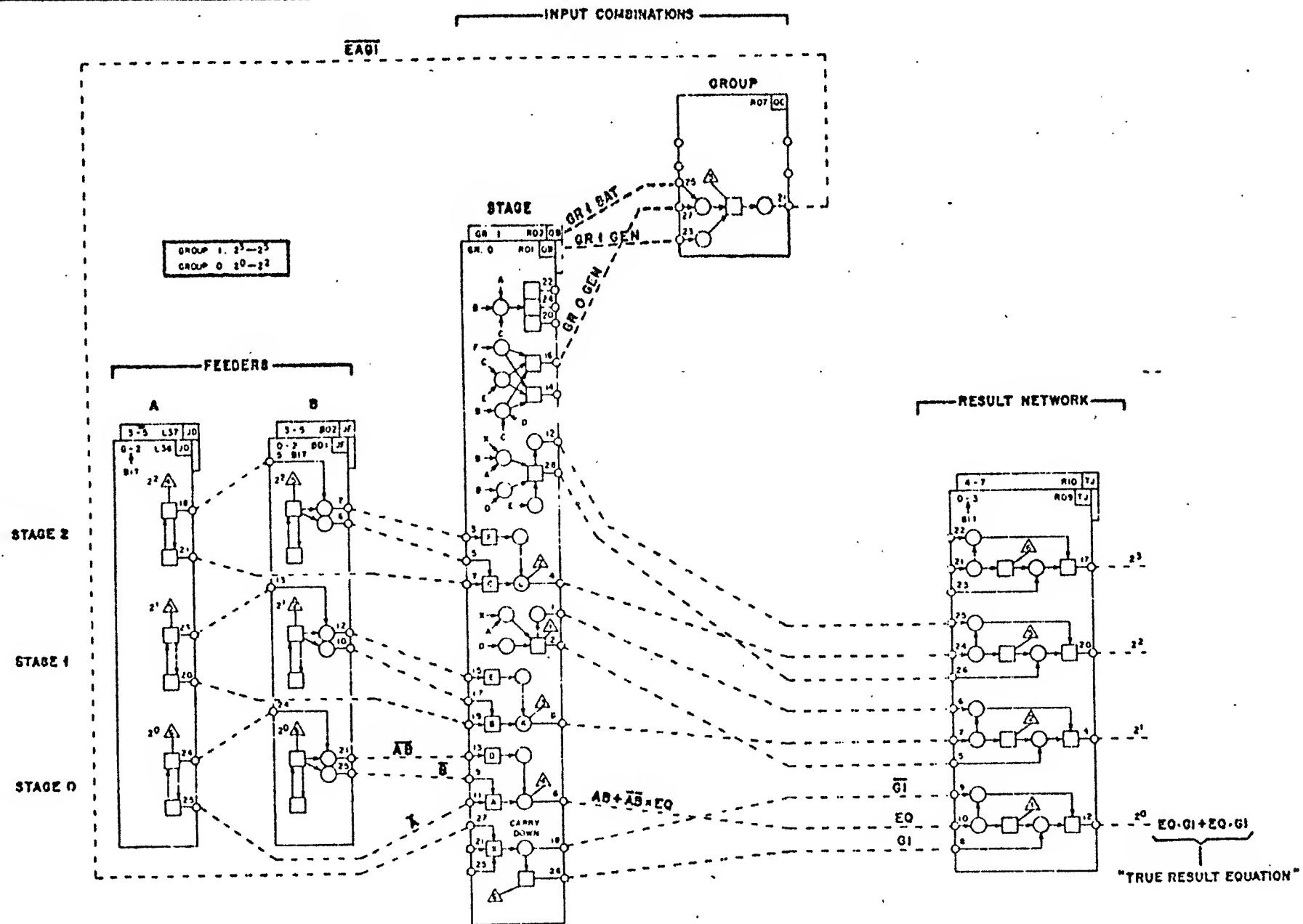
$+$  = OR

NO.	STATEMENT
1	$A = 0 \neq 1$ or $A = 1$ if $A \neq 0$
2	$0 \neq 1, 1 \neq 0$
3	$0 \cdot 0 = 0, 0 + 0 = 0$
4	$1 \cdot 1 = 1, 1 + 1 = 1$
5	$0 \cdot 1 = 0, 0 + 1 = 1$
6	$A + A = A$
7	$A \cdot A = A$
8	$A + 1 = 1$
9	$A \cdot 0 = 0$
10	$A + 0 = A$
11	$A \cdot 1 = A$
12	$A + \bar{A} = 1$
13	$A \cdot \bar{A} = 0$
14	$\bar{\bar{A}} = A$
15	$A + AB = A$
16	$A(B+C) = AB+AC, A+(B \cdot C) = (A+B)(A+C)$
17	$A + \bar{A}B = A+B, A(\bar{A}+B) = A \cdot B$
18	$\overline{A+B} = \bar{A} \cdot \bar{B}, \overline{A \cdot B} = \bar{A} + \bar{B}$

"YOU MAY BREAK THE LINE IF YOU CHANGE THE SIGN."

# INCREMENTER MODULE





CONTROL DATA  
CORPORATION  
COMPUTER DIVISION

CENTRAL PROCESSOR  
ADDER

6601/04  
60119300  
FIG 13

$$\begin{array}{r}
 \text{MULTIPLICAND (Md)} \\
 \times \text{MULTIPLIER (Mr)} \\
 \hline
 \text{PRODUCT (Pr)}
 \end{array}$$

### MULTIPLY ALGORITHM

1. Convert Mr and Md to positive operands. (Note original signs.)
2. Check the LSD of the Mr if it is;
  - "1" = Add the Md to the Pr
  - "0" = Add zero's to the Pr (do Nothing)
3. Left shift the Md (shift a zero into the LSD).
4. Right shift the Mr (End Off).
5. Repeat steps 2 through 4 until the Mr is reduced to nothing. At this point check the signs of the original Md and Mr if;
  - = Pr is the correct result
  - ≠ Complement Pr to obtain the correct result.

### EXAMPLES

$$5_{10} \times 5_{10} = 25_{10}$$

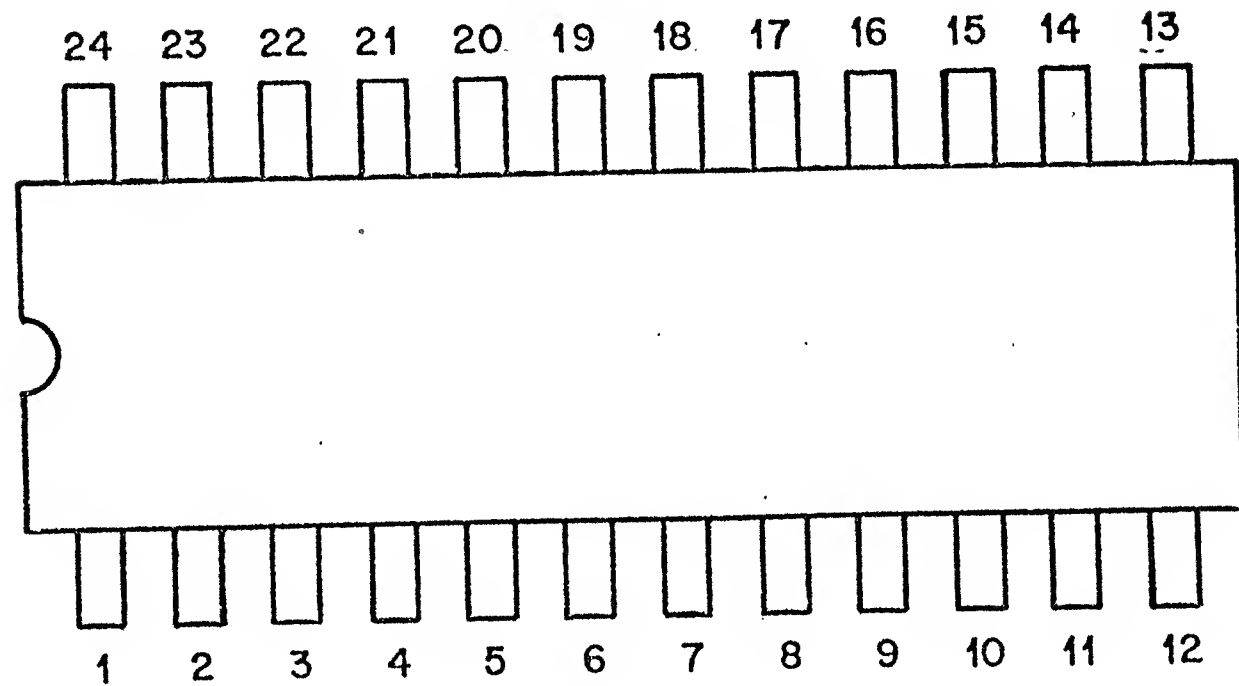
	Md	Mr	Pr
Start	000101	000101	000000
Step 1	000101	000101	000000
Step 2		000101	000000 + 000101 000101
Step 3-4	001010	00010	-
Step 2	-	00010	000101
Step 3-4	010100	0001	-
Step 2	-	0001	000101 + 010100 011001
Step 3-4	101000		
Step 5			011001

$$-7_{10} \times 6_{10} = -42_{10}$$

	Md	Mr	Pr
Start	0110	1000	0000
Step 1	110	111	000
Step 2	-	111	000 +110 110
Step 3-4	1100	11	-
Step 2		11	110 +1100 10010
Step 3-4	11000	1	-
Step 2	-	1	10010 +11000 101010
Step 3-4	110000	( )	-
Step 5	0101010	→	1010101

# ARITHMETIC AND LOGIC UNIT

## CHIP PIN LAYOUT



# DIVIDE ALGORYTHM CONT

Step	Divisor	Remainder	Quotient
4	{Same}	0 0 0 1 0 1 0 <u>0</u>	{Same}
3b	{Same}	<div> 0 0 0 1 0 1 0 0  -       0 1 0 1 0 0  <hr/> 0 0 0 0 0 0 0 0 </div>	0 1.0 1 -
5-a,b	{Same}	- - -	0 1 0 1. {LS-2}

## VOLTAGE LEVELS

LOGIC LOGIC TYPE	HIGH	LOW
ECL	-0.8V (LOGICAL 0)	-1.6V (LOGICAL 1)
TTL	+3.6V (LOGICAL 1)	+0.8V (LOGICAL 0)
MOS	+20.0V (LOGICAL 1)	GROUND (LOGICAL 0)
POWER SUPPLY LOGIC POWER		
ECL	GROUND, -2.2V AND -5.2 VOLTS	
TTL	+5 AND +8 VOLTS	
MEMORY	+20V AND +22.5V	

# KEY TO LOGIC SYMBOLS

## SYMBOLS

## LOGIC FUNCTION

$\&$

AND GATE

$\vee$

OR GATE / INVERTER

$\oplus$

EXCLUSIVE OR

00

ODD PARITY (DUE TO INVERSION)

$X \rightarrow Y$

ENCODER OR DECODER

$X/Y$

LOGIC LEVEL CONVERTER

RGTR

REGISTER

ALU

ARITHMETIC LOGIC UNIT

RCVR

RECEIVER

MUX

MULTIPLEXER

L

"1" = LOW = MOST NEGATIVE SIGNAL

H

"0" = HIGH = MORE POSITIVE SIGNAL

OPEN INPUT PIN

LOGIC "1" INPUT = LOW

GROUND INPUT PIN

LOGIC "0" INPUT = HIGH

SR

SHIFT REGISTER

CNTR

COUNTER

ENDCD

ENCODER

# KEY TO LOGIC SYMBOLS

## DESIGNATOR

## DESCRIPTION



LOW INPUT OR OUTPUT



GROUPED INPUTS

C

GATING (CLOCK) INPUT

G

GATING INPUT THAT AFFECT  
OTHER INPUT OR OUTPUTS

G1, G2

RELATIVE WEIGHTING OF INPUTS  
OR OUTPUTS IN CODES

S

SET INPUT

R

RESET INPUT

D

DATA INPUT OF D TYPE F/F

GS

GATED INPUT SET

GR

GATED INPUT RESET

# KEY TO LOGIC SYMBOLS

## DESIGNATOR

## DESCRIPTION



INDICATES THE GATING ACTION OCCURES ON THE TRAILING EDGE OF THE GATING INPUT.

INDICATES THE GATING ACTION OCCURES ON THE LEADING EDGE OF THE GATING INPUT.



RIGHT SHIFT OR SHIFT DOWN.



LEFT SHIFT OR SHIFT UP.



WIRED AND GATE



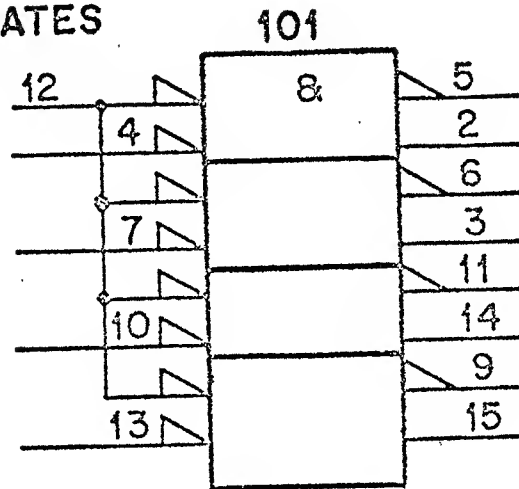
WIRE CONNECTION



WIRE CROSSOVER

# KEY TO LOGIC SYMBOLS-ECL

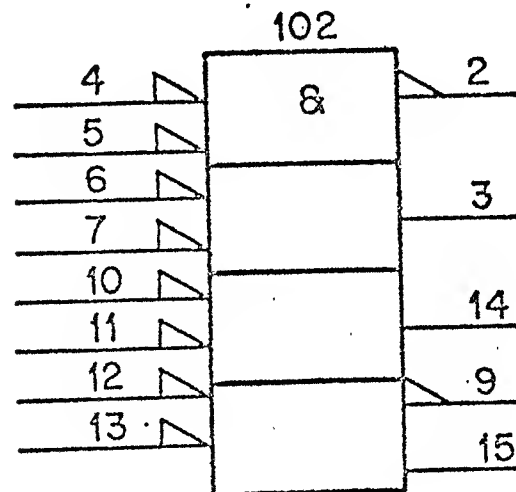
## AND GATES



2 INPUT QUAD AND  
PIN 12 USED AS  
CONTROL

## TRUTH TABLE

A	B	PIN 5	PIN 2
H	H	H	L
L	H	H	L
H	L	H	L
L	L	L	H

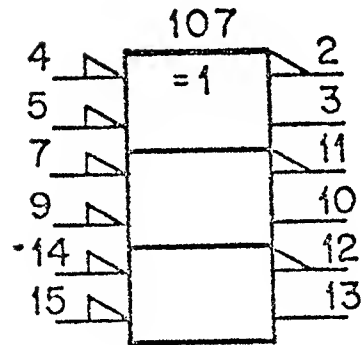


QUAD 2 INPUT  
AND

## TRUTH TABLE

PIN 12	PIN 13	PIN 9	PIN 15
H	H	H	L
L	H	H	L
H	L	H	L
L	L	L	H

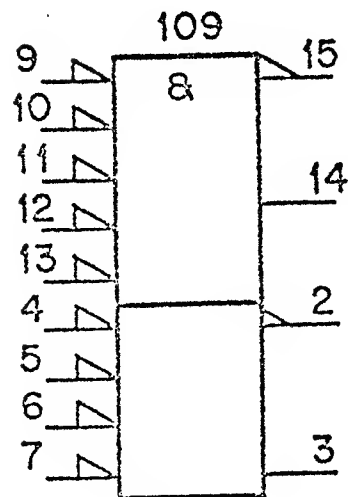
# KEY TO LOGIC SYMBOLS - ECL



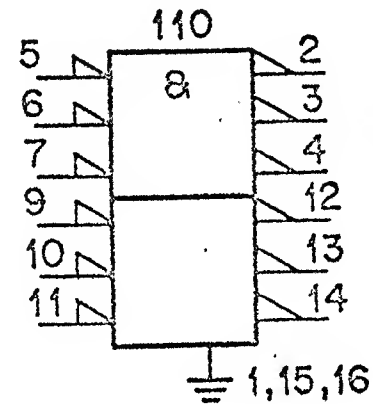
EXCLUSIVE OR

TRUTH TABLE

PIN 4	PIN 5	PIN 2	PIN 3
H	H	H	L
H	L	L	H
L	H	L	H
L	L	H	L

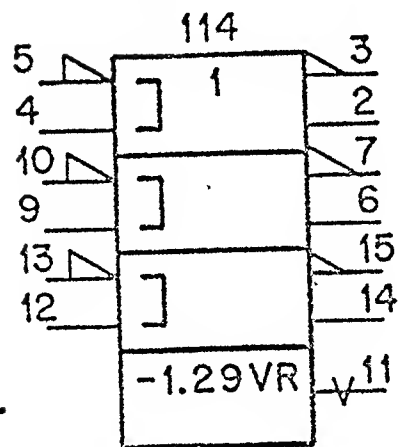


DUAL 4-5  
INPUT  
AND/NAND



DUAL  
3 INPUT/3 OUTPUT  
AND

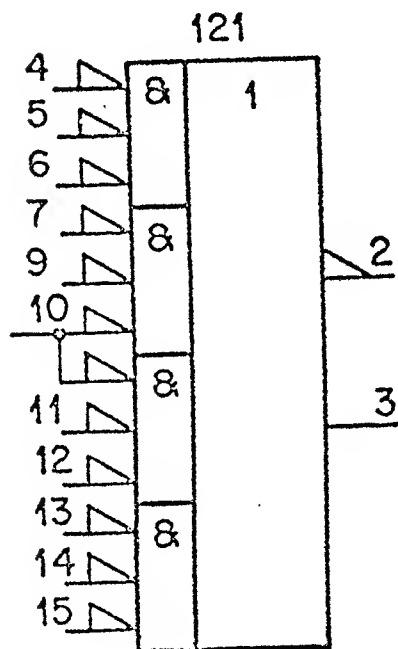
# KEY TO LOGIC SYMBOLS - ECL



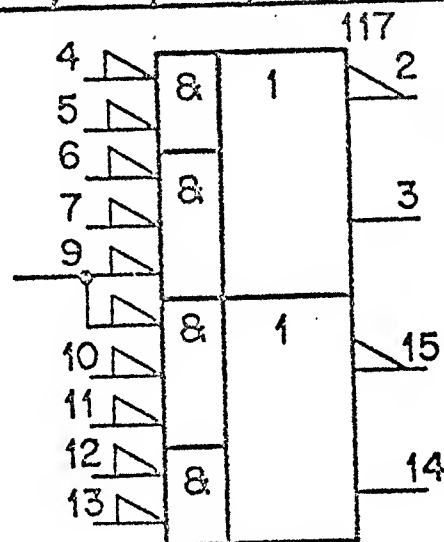
TWISTED PAIR  
OR  
COAX RECEIVER

TRUTH TABLE 117 CHIP

PIN4	PIN5	PIN6	PIN7	PIN9	PIN2	PIN3
H	H	H	H	H	H	L
L	H	H	H	H	H	L
H	L	H	H	H	H	L
H	H	L	H	H	H	L
H	H	H	L	H	H	L
H	H	H	H	L	H	L
L	L	H	H	H	L	H

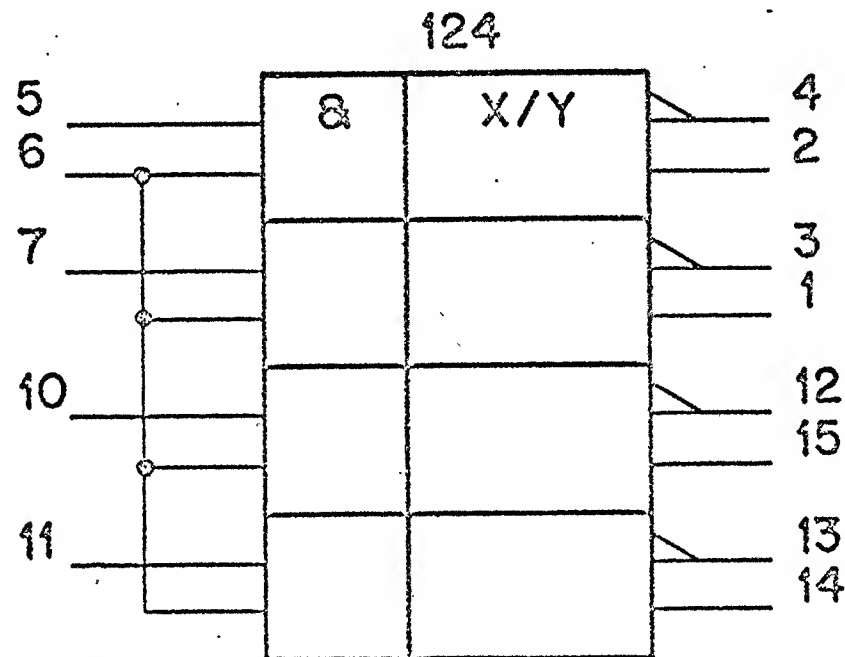


4 WIDE AND-ØR



DUAL AND -  
ØR

# KEY TO LOGIC SYMBOLS-ECL

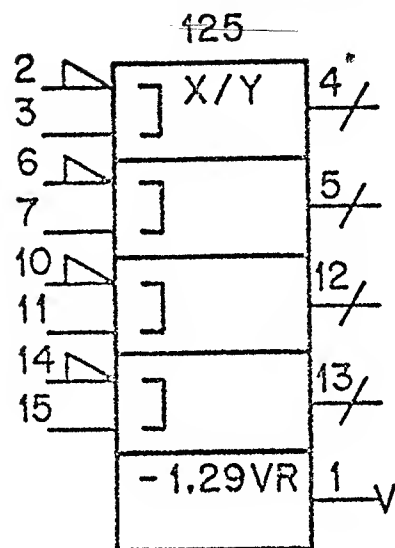


QUAD TTL TO ECL TRANSLATOR

## 124 TTL/ECL

- 1) PIN 6 = GATE = HIGH TO ALLOW PASSAGE (NORMALLY KEPT AT +5V)  
= LOW, ALL "1" = OUTPUTS = 1. (ALL "0" OUTPUTS = 0)
- 2) INPUT 1 = +3.6V. OUTPUT 1 = -1.6V.  
INPUT 0 = +0.8V OUTPUT 0 = -0.8V

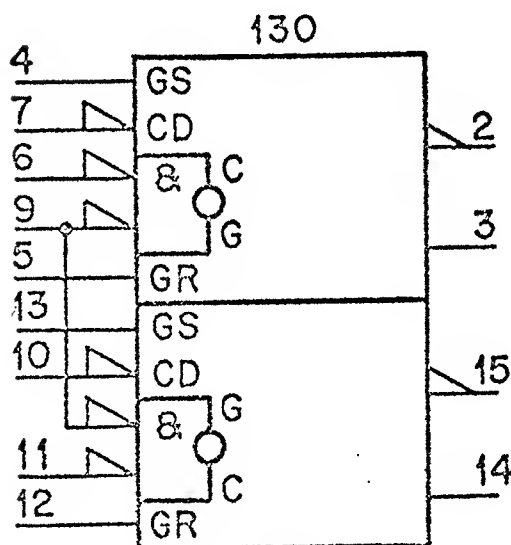
# KEY TO LOGIC SYMBOLS-ECL



## QUAD ECL TO TTL TRANSLATOR

### 125 ECL/TTL

- 1) PIN 1 CONNECTED TO EVEN PINS = INVERSION. PIN 1 CONNECTED TO ODD PINS = INVERSION. ECL INPUT 1 = -1.6V. TTL OUTPUT = +3.6V. ECL INPUT 0 = -0.8V, TTL OUTPUT = +0.8V

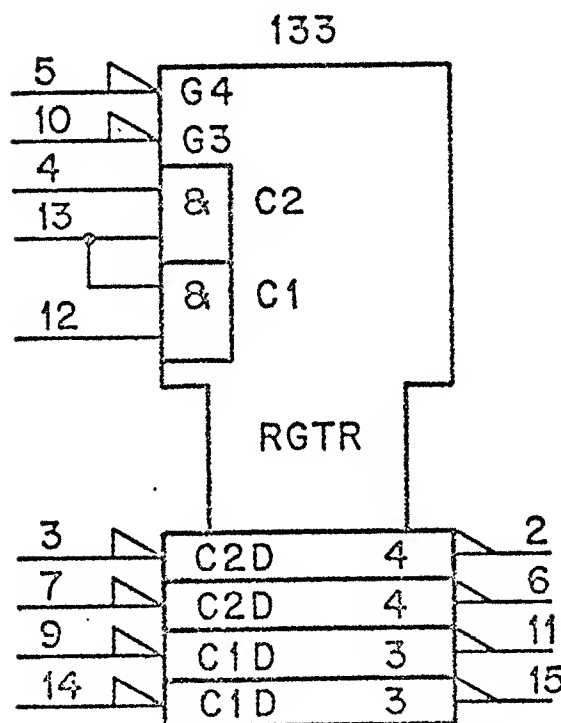


## DUAL LATCH

- PIN 9 = COMMON GATE
- PINS 6, 11 = CLOCK INPUTS
- PINS 7, 10 = CLOCKED DATA
- PINS 4, 13 = FORCE SET
- PINS 5, 12 = FORCE RESET

- 1) LATCHES ON  $\overline{A}$  (ON TRAILING EDGE OF CLOCK).
- 2) FORCE SET/CLEAR WORKS WHEN CLOCK (HIGH).
- 3) O/P FOLLOWS I/P WHEN CLOCK (LOW).

## KEY TO LOGIC SYMBOLS-ECL



### QUAD LATCH

PIN 13 = COMMON GATE

PINS 4,12 = CLOCK INPUTS

PINS 3,7,9,14 = CLOCKED DATA

PINS 5,10 = OUTPUT GATES

PINS 4 AND 13 CLOCK DATA IN FOR PINS 3 AND 7.

PINS 12 AND 13 CLOCK DATA IN FOR PINS 9 AND 14.

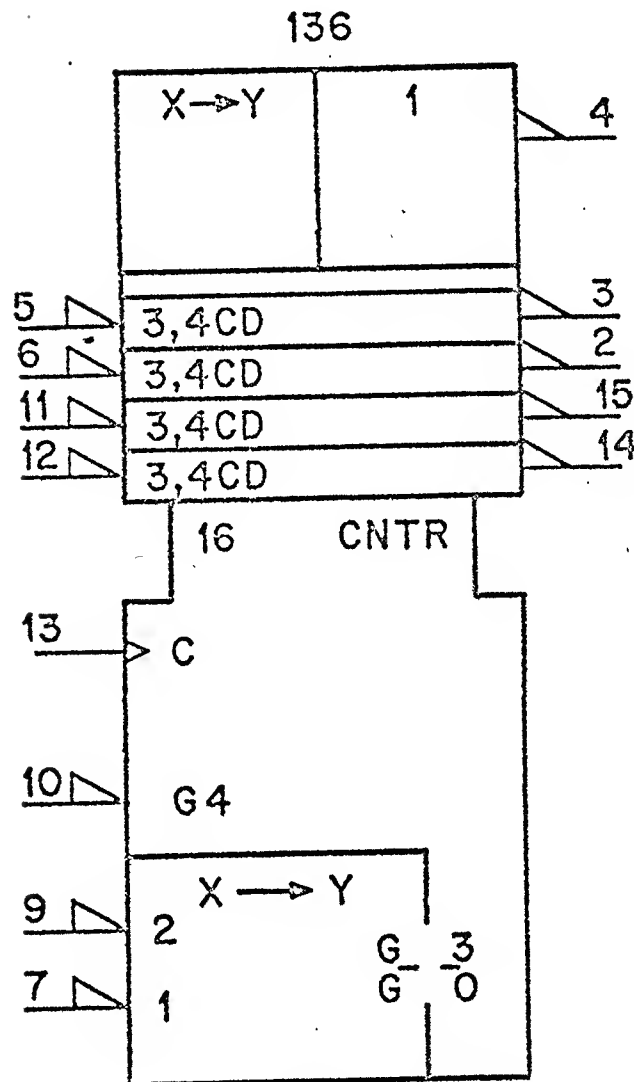
PIN 5 GATES DATA OUT FOR PINS 2 AND 6.


PIN 10 GATES DATA OUT FOR PINS 11 AND 15.

- 1) LATCHES ON  $\overline{1}$  (ON LEADING EDGE) } OPPOSITE OF 130.
- 2) O/P FOLLOWS I/P WHEN CLOCK (HIGH) }
- 3) O/P IS GATED (PIN 5,10), WHEN GATE = HIGH, ALL O/P'S = 1

# KEY TO LOGIC SYMBOLS - ECL

## UNIVERSAL HEXADECIMAL COUNTER



1) LATCHES ON  (ON TRAILING EDGE)

2) PIN 9. PIN 7  
 LOW LOW = PRESET PARALLEL  
 HIGH LOW = -1 INPUTS  
 LOW HIGH = +1  
 HIGH HIGH = STOP COUNTING

3) PIN 10 INHIBITS CLOCK INPUTS AND CARRY OUTPUT WHEN HIGH.

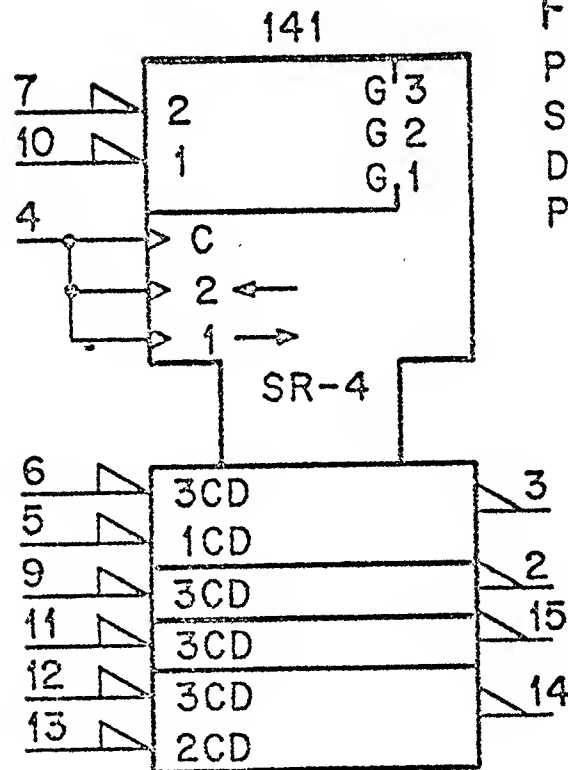
4) CARRY OUTPUT (PIN 4) = LOW WHEN COUNT IS "1111" WHEN INCREMENTING AND "0000" WHEN DECREMENTING

PINS 5, 6, 11, 12 = CLOCK DATA

PINS 10 = CARRY IN

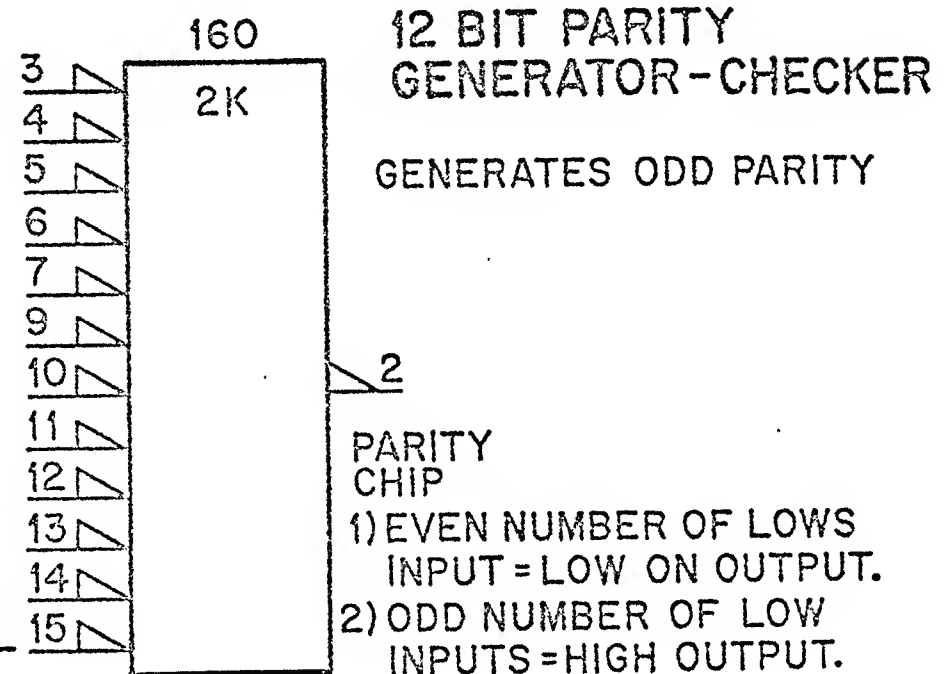
INCREMENTS OR DECREMENTS AT A RATE DETERMINED BY THE CLOCK ON PIN 13

# KEY TO LOGIC SYMBOLS-ECL



## FOUR BIT UNIVERSAL SHIFT REGISTER

PINS 6,9,11,12 = CLOCKED DATA  
SHIFTING IS DONE AT A RATE  
DETERMINED BY THE CLOCK ON  
PIN 4



## 12 BIT PARITY GENERATOR-CHECKER

GENERATES ODD PARITY

PARITY  
CHIP

- 1) EVEN NUMBER OF LOWS  
INPUT = LOW ON OUTPUT.
- 2) ODD NUMBER OF LOW  
INPUTS = HIGH OUTPUT.

## 141 SHIFTER

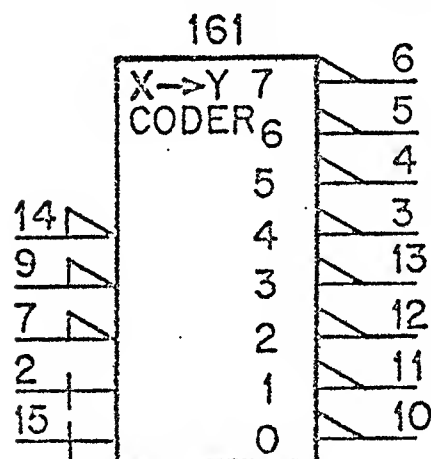
- 1) LATCHES ON  $\bar{5}$  (ON TRAILING EDGE)
- 2) PIN 10 PIN 7  
LOW LOW = PRESET PARALLEL  
INPUTS  
LOW HIGH = SHIFT RIGHT  
HIGH LOW = SHIFT LEFT
- 3) PIN 5 = INPUT FOR RIGHT SHIFT (N+4)  
PIN 13 = INPUT FOR LEFT SHIFT (N-1)

# KEY TO LOGIC SYMBOLS-ECL

TRUTH TABLE

ENABLE INPUTS		INPUTS			OUTPUTS							
PIN 15	PIN 2	PIN 14	PIN 9	PIN 7	PIN 6	PIN 5	PIN 4	PIN 3	PIN 13	PIN 12	PIN 11	PIN 10
L	L	L	L	L	L	H	H	H	H	H	H	H
L	L	L	L	H	H	L	H	H	H	H	H	H
L	L	L	H	L	H	H	L	H	H	H	H	H
L	L	L	H	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	L	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	Ø	Ø	Ø	H	H	H	H	H	H	H	H
L	H	Ø	Ø	Ø	H	H	H	H	H	H	H	H
H	H	Ø	Ø	Ø	H	H	H	H	H	H	H	H

Ø = DON'T CARE



## BINARY 1-8 LOW DECODER

PINS 7,9,14 = CODED INPUT

DECODER

1) PIN 2 AND 15 MUST BE BOTH LOW.

IF EITHER HIGH, OUTPUTS = HIGH.

2) ONLY ONE OUTPUT WILL BE LOW, FOR ANY PATTERN INPUT.

# KEY TO LOGIC SYMBOLS-ECL

## 8 LINE TO 1 LINE MULTIPLEXER TRUTH TABLE

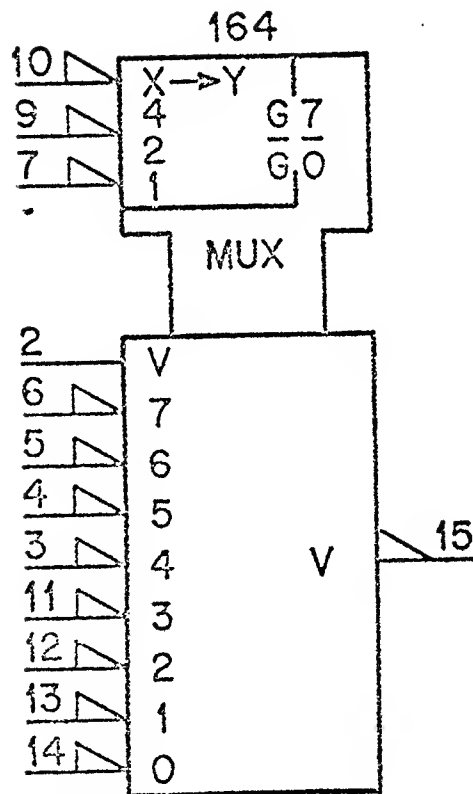
PIN 2	ADDRESS INPUTS			OUTPUT ON PIN 15
	PIN 10	PIN 9	PIN 7	
L	L	L	L	PIN 6
L	L	L	H	PIN 5
L	L	H	L	PIN 4
L	L	H	H	PIN 3
L	H	L	L	PIN 11
L	H	L	H	PIN 12
L	H	H	L	PIN 13
L	H	H	H	PIN 14
H	Ø	Ø	Ø	L

Ø = DON'T CARE

PINS 7,9,10 = CONTROL PINS

PIN 2 = ENABLE

PINS 6,5,4,3,11,12,13,14 - INPUT PINS



### 164 MULTIPLEXER

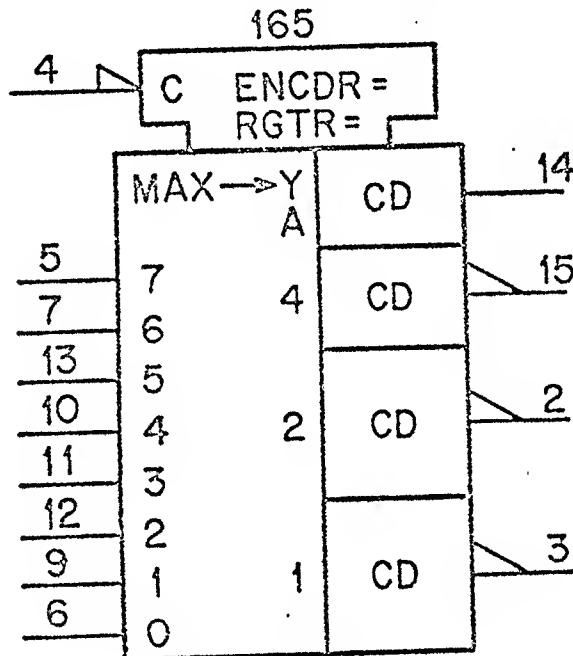
- 1) PIN 2 MUST BE LOW. IF HIGH, OUTPUT IS FORCED LOW.
- 2) PATTERN ON THE 3 INPUT CONTROL PINS, DECIDE WHICH INPUT ALLOWS TO OUTPUT.

# KEY TO LOGIC SYMBOLS-ECL


TRUTH TABLE

DATA INPUTS								OUTPUTS			
PIN 5	PIN 7	PIN 13	PIN 10	PIN 11	PIN 12	PIN 9	PIN 6	PIN 14	PIN 15	PIN 2	PIN 3
H	Ø	Ø	Ø	Ø	Ø	Ø	Ø	H	L	L	L
L	H	Ø	Ø	Ø	Ø	Ø	Ø	H	L	L	H
L	L	H	Ø	Ø	Ø	Ø	Ø	H	L	H	L
L	L	L	H	Ø	Ø	Ø	Ø	H	L	H	H
L	L	L	L	H	Ø	Ø	Ø	H	H	L	L
L	L	L	L	L	H	Ø	Ø	H	H	L	H
L	L	L	L	L	L	H	Ø	H	H	H	L
L	L	L	L	L	L	L	H	H	H	H	H
L	L	L	L	L	L	L	L	L	L	L	L

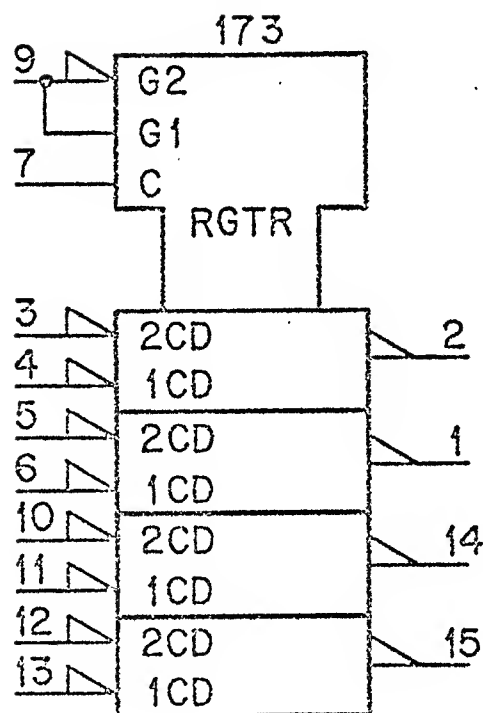
Ø = DON'T CARE



## 8 INPUT PRIORITY ENCODER


LATCHES ON  (ON TRAILING EDGE)  
 OUTPUT FOLLOWS INPUT WHEN CLOCK IS LOW  
 PIN 14 HIGH WHEN ANY INPUT HIGH  
 PIN 4 = CLOCK

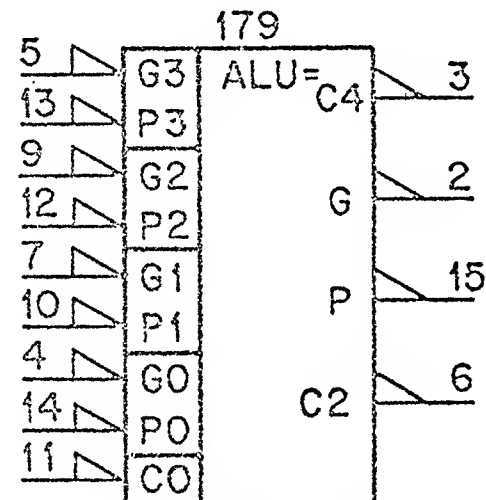
# KEY TO LOGIC SYMBOLS-ECL



QUAD 2 TO 1 MULTIPLEXER LATCH  
 PIN 9 = GATE PIN 7 = CLOCK  
 CLOCK AND G1 GATES 1CD INPUTS  
 CLOCK AND G2 GATES 2CD INPUTS

LOOK AHEAD CARRY BLOCK  
 C2 = GROUP 2 CARRY IN.  
 P = ALL GROUP PASSES.  
 G = CARRY FROM GROUP 3 EXCLUDING CO.  
 CO = CARRY INTO GROUPS,  
 FROM ANOTHER 179 CHIP.  
 C4 = CARRY FROM GROUP 3 INCLUDING CO.

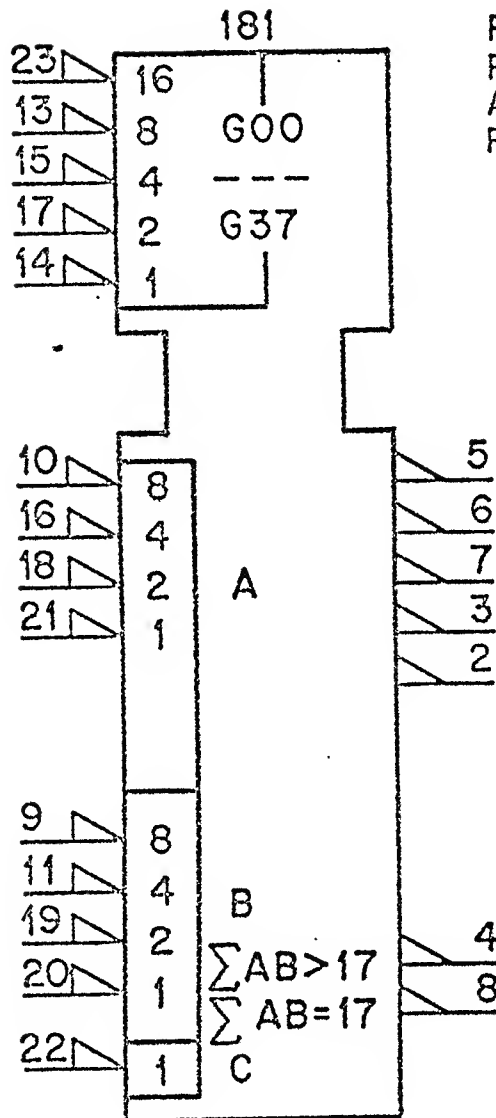
- 1) LATCHES ON  (ON TRAILING EDGE)
- 2) O/P FOLLOWS I/P WHEN CLOCK (LOW)
- 3) GATE SELECTS I/P. PIN 9=0=1D I/P.  
 (PIN 9=1, 2D I/P)



# KEY TO LOGIC SYMBOLS-ECL

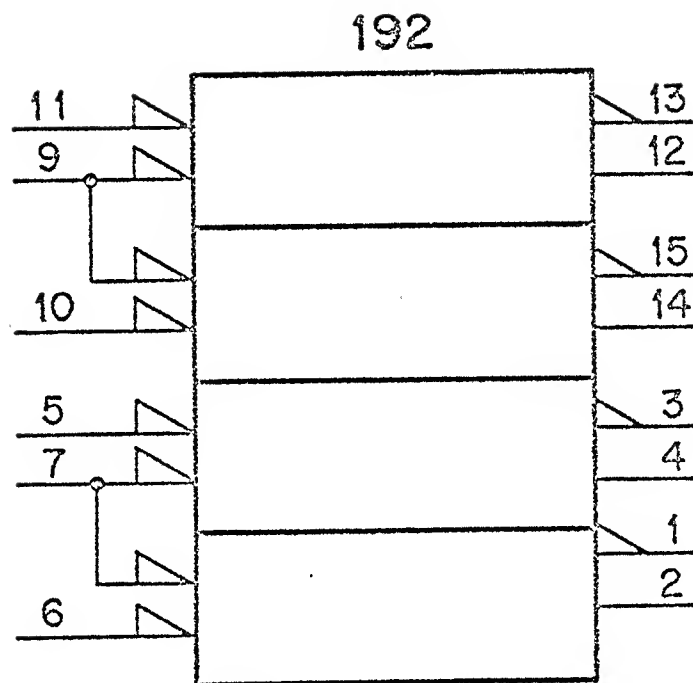
## 4 BIT ARITHMETIC LOGIC UNIT

PIN 22=GROUP CARRY IN. PIN 4=GROUP CARRY OUT EXCLUDING C  
 PIN 8=GROUP PASS. PINS 6,7,3,2=DATA OUT.  
 A AND B=DATA IN. PIN 5=GROUP CARRY OUT INCLUDING C.  
 PINS 23,13,15,17,14 CONTROL ALU FUNCTION (SEE TABLE)



FUNCTION SELECT PINS 13, 15, 17, 14	LOGIC FUNCTIONS 23=M IS HIGH	ARITHMETIC OPERATION PIN 23=M IS LOW
	F	F
L L L L	$F = \bar{A}$	$F = A \text{ MINUS } 1$
L L L H	$F = \bar{A} + \bar{B}$	$F = A \text{ PLUS } (A + \bar{B})$
L L H L	$F = \bar{A} \cdot B$	$F = A \text{ PLUS } (A + B)$
L L H H	$F = \text{LOGICAL "0"}$	$F = A \text{ TIMES } 2$
L H L L	$F = \bar{A} \cdot \bar{B}$	$F = (A \cdot B) \text{ MINUS } 1$
L H L H	$F = \bar{B}$	$F = (A \cdot B) \text{ PLUS } (A + \bar{B})$
L H H L	$F = A \oplus B$	$F = A \text{ PLUS } B$
L H H H	$F = A \cdot \bar{B}$	$F = A \text{ PLUS } (A \cdot B)$
H L L L	$F = \bar{A} + B$	$F = (A \cdot \bar{B}) \text{ MINUS } 1$
H L L H	$F = A \oplus \bar{B}$	$F = A \text{ MINUS } B \text{ MINUS } 1$
H L H L	$F = B$	$F = (A \cdot \bar{B}) \text{ PLUS } (A + B)$
H L H H	$F = A \cdot B$	$F = (A \cdot \bar{B}) \text{ PLUS } A$
H H L L	$F = \text{LOGICAL "1"}$	$F = \text{MINUS } 1 \text{ (TWO'S COMPLEMENT)}$
H H L H	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 0$
H H H L	$F = A + B$	$F = (A + B) \text{ PLUS } 0$
H H H H	$F = A$	$F = A \text{ PLUS } 0$

## KEY TO LOGIC SYMBOLS-ECL

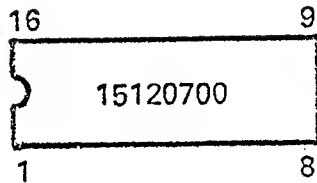
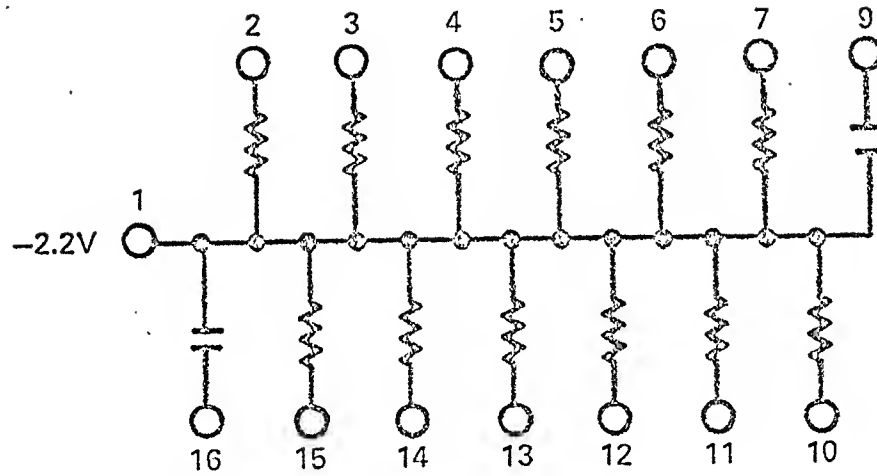


TWISTED PAIR OR  
COAX TRANSMITTER

L7000

HO-27

# R100 TERMINATORS



PIN 1 = -2.2 VOLTS

PINS 9 AND 16 = GROUND

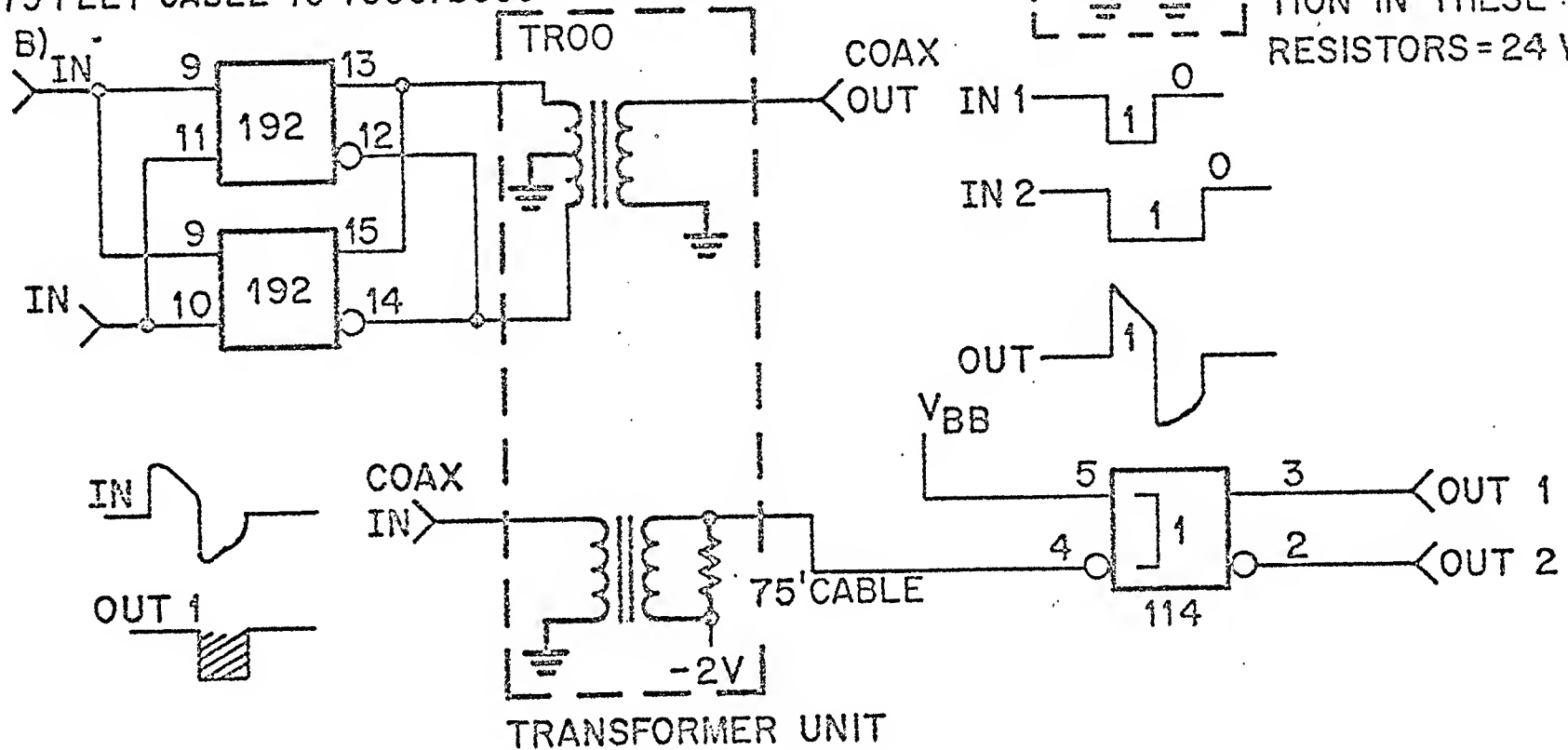
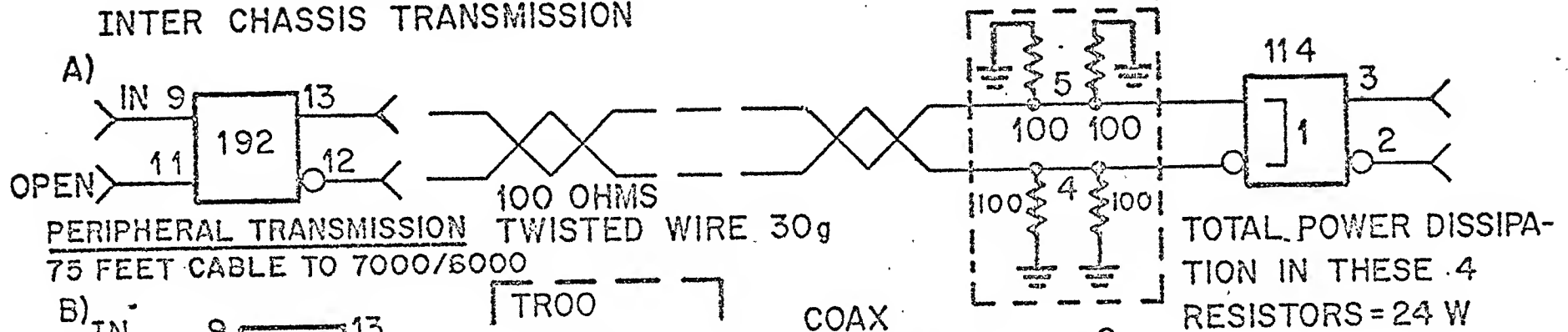
PINS 1 THRU 7 AND 9 THRU 15 = 100 OHMS TO -2.2 VOLTS

PIN 8 = OPEN

CAPACITOR = .01 MICROFARAD

# INTERFACES USING 10192

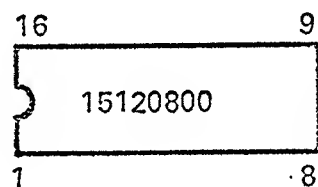
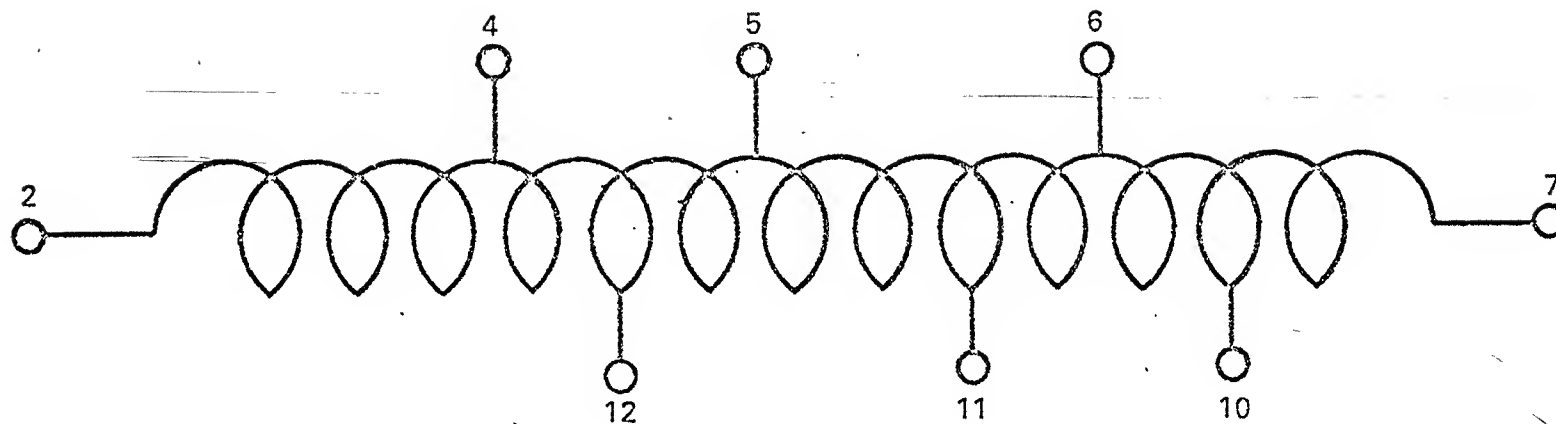
## INTER CHASSIS TRANSMISSION



L7000

HO-26

# DL17 DELAY



## PINS

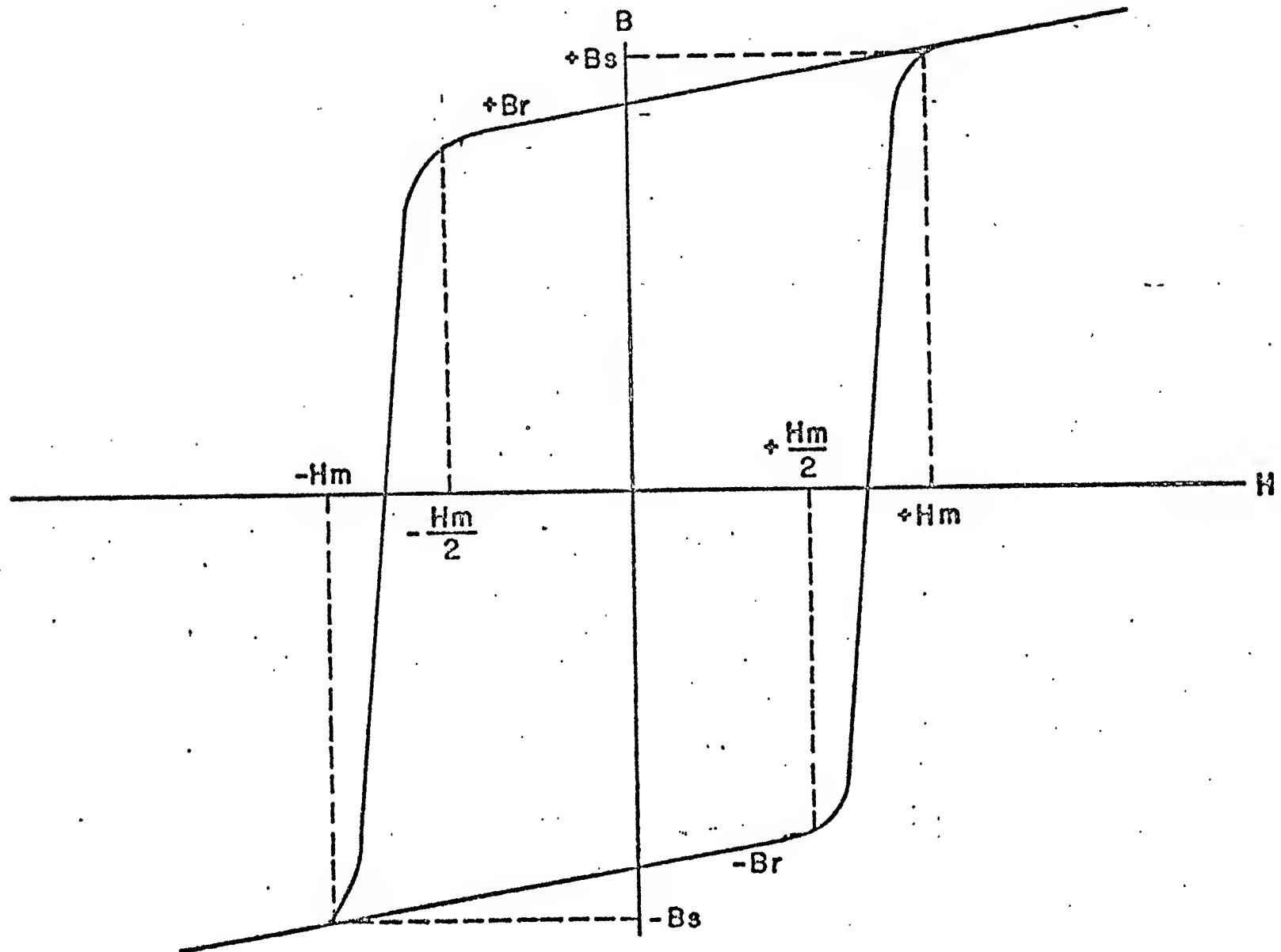
2=INPUT  
 7=TERMINATED  
 4=8.5 NSEC  
 12=10.0 NSEC  
 5=11.5 NSEC  
 11=13.0 NSEC  
 6=14.5 NSEC  
 10=16.0 NSEC  
 7=17.5 NSEC

## PINS

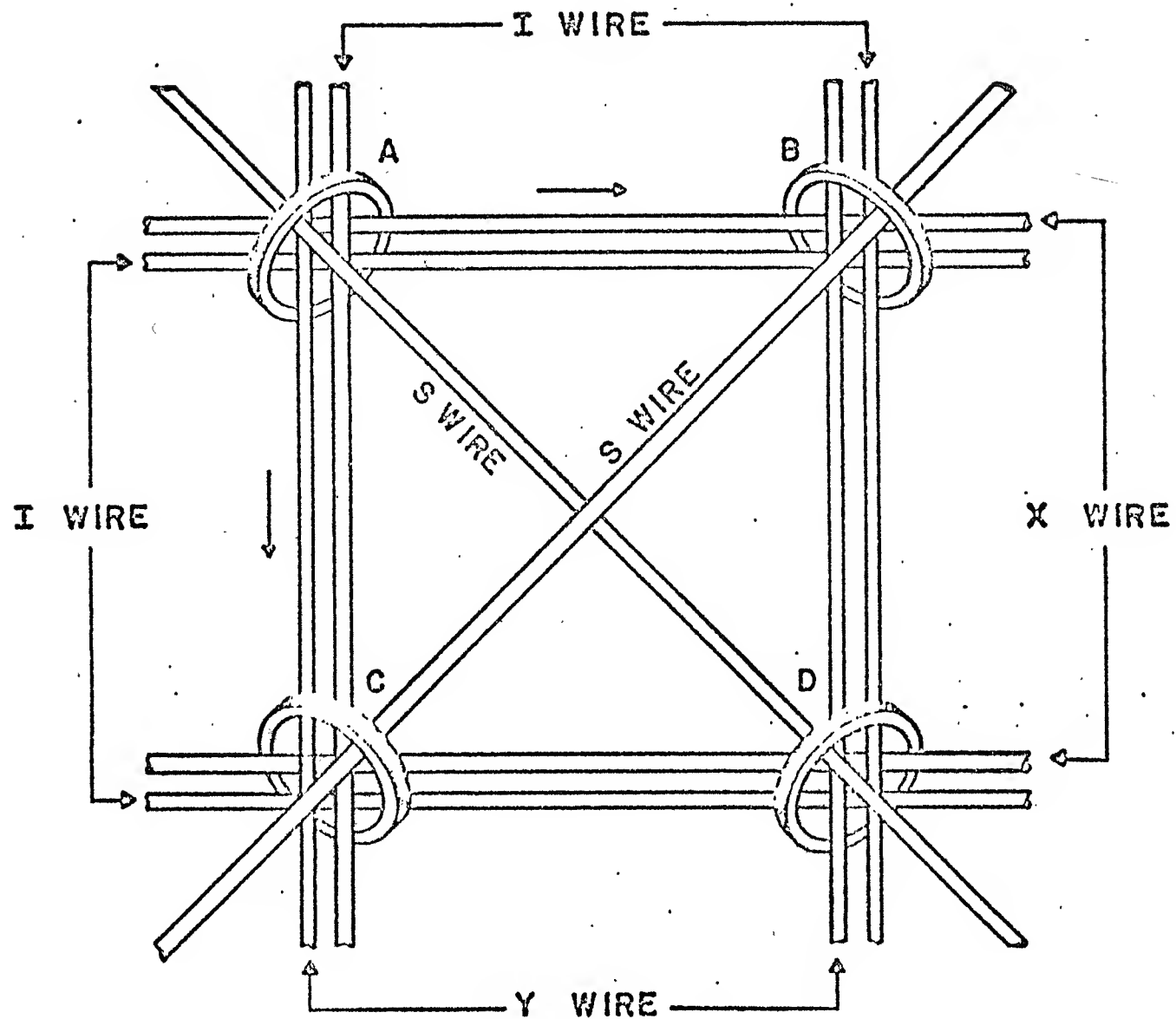
7=INPUT  
 2=TERMINATED  
 10=1.5 NSEC  
 6=3.0 NSEC  
 11=4.5 NSEC  
 5=6.0 NSEC  
 12=7.5 NSEC  
 4=9.0 NSEC  
 2=17.5 NSEC

PIN 16= GROUND

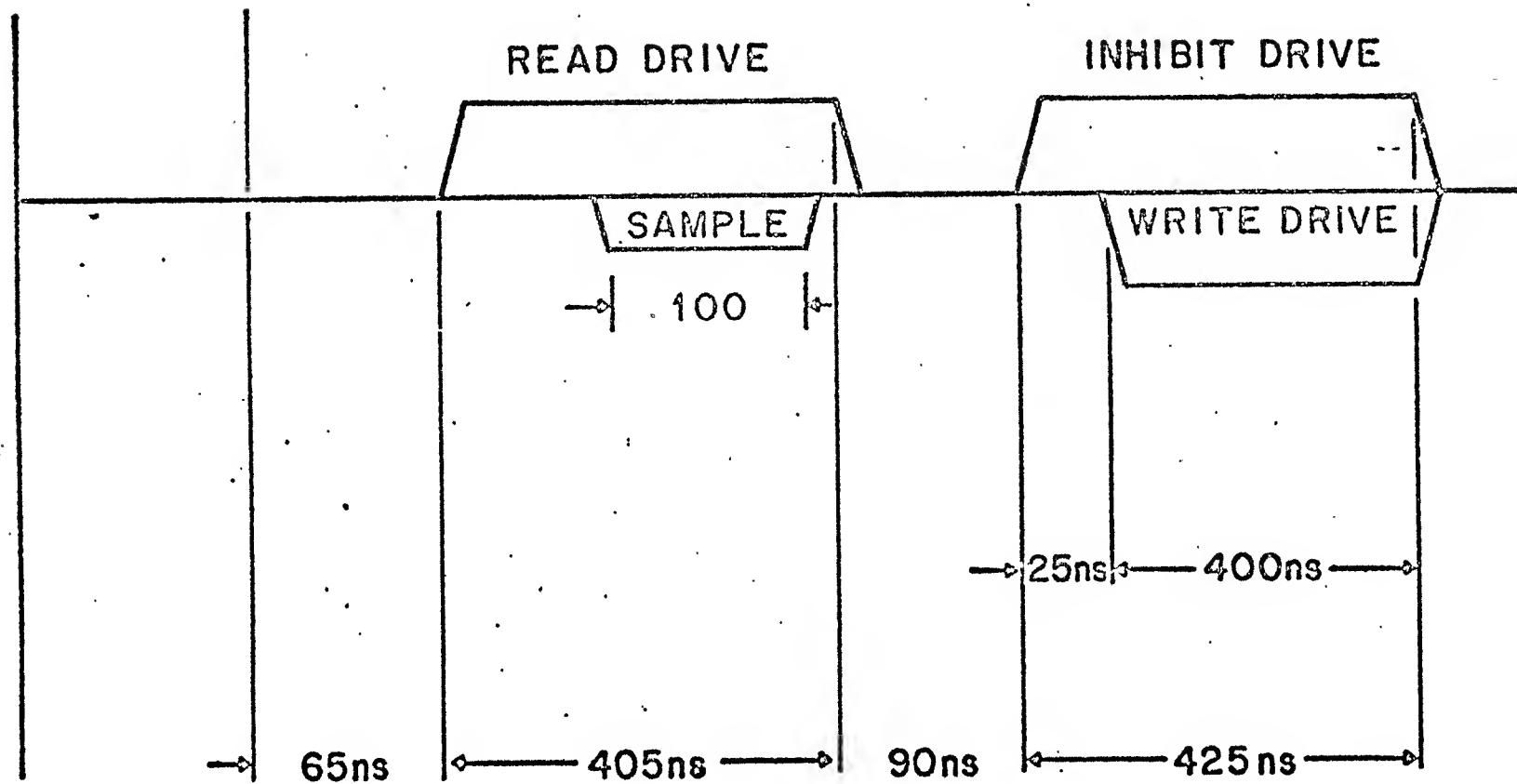
# HYSTERESIS LOOP



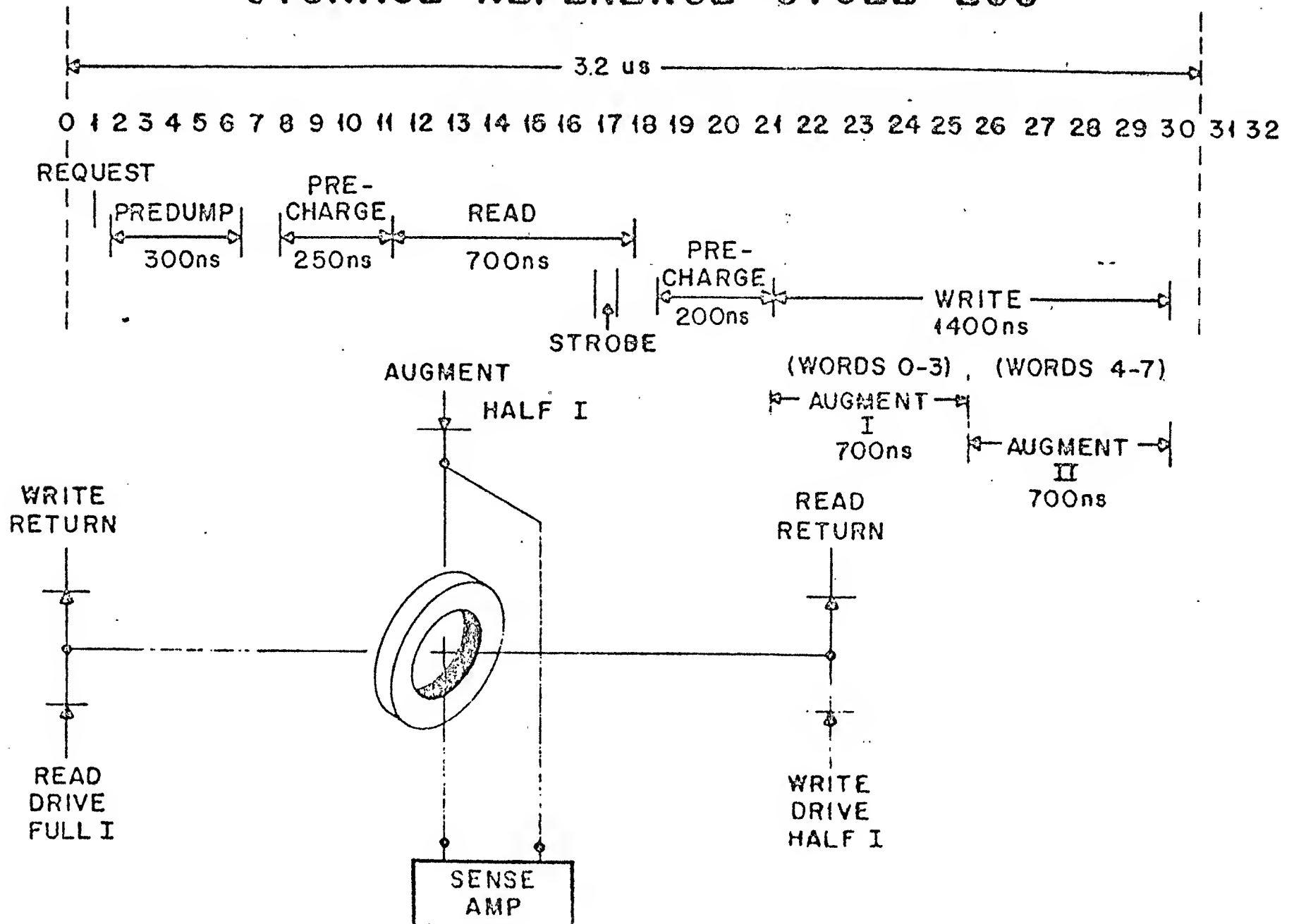
# CORE SELECTION



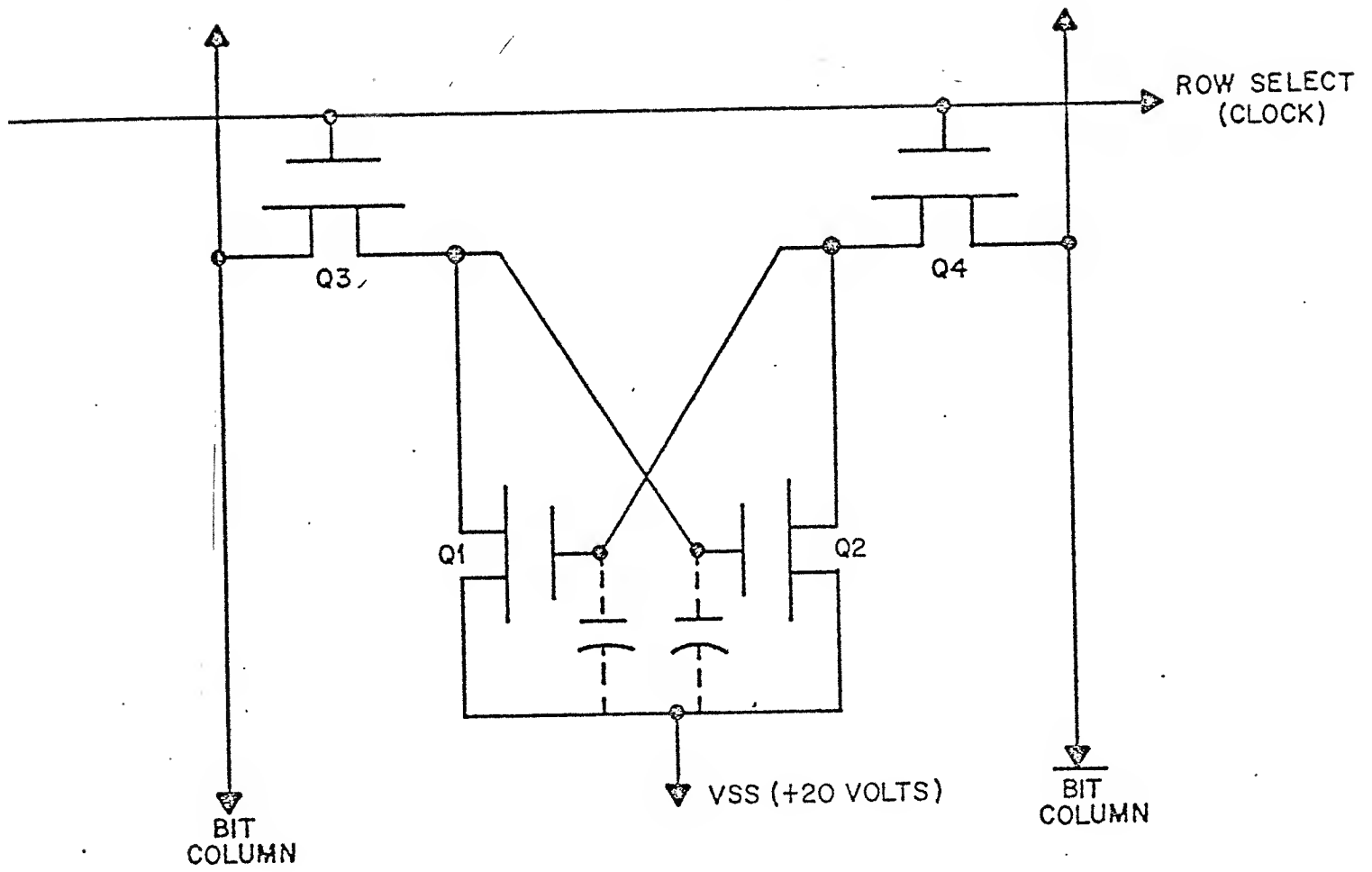
# MEMORY CYCLE TIMING



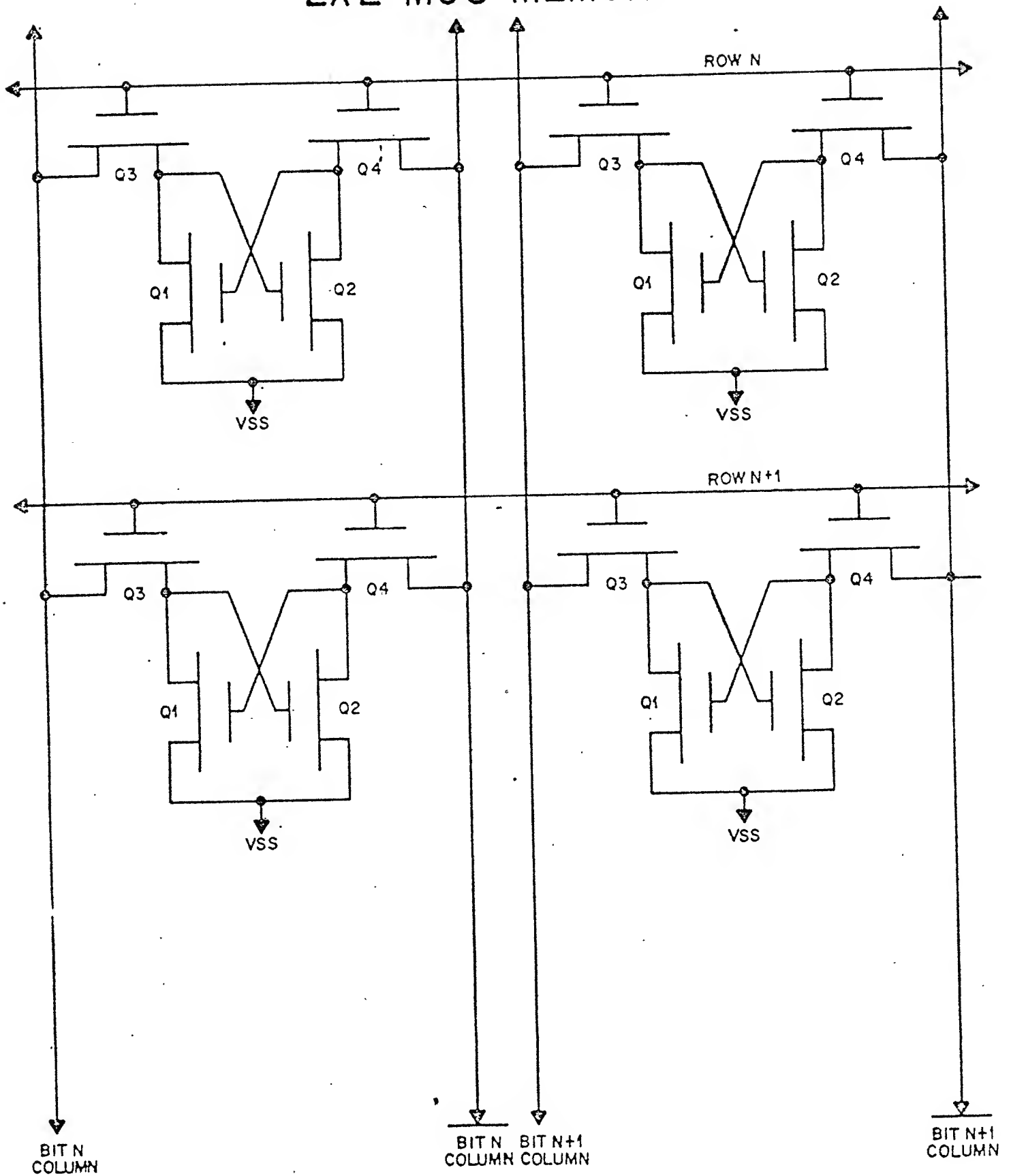
# STORAGE REFERENCE CYCLE ECS



# ONE MOS CELL



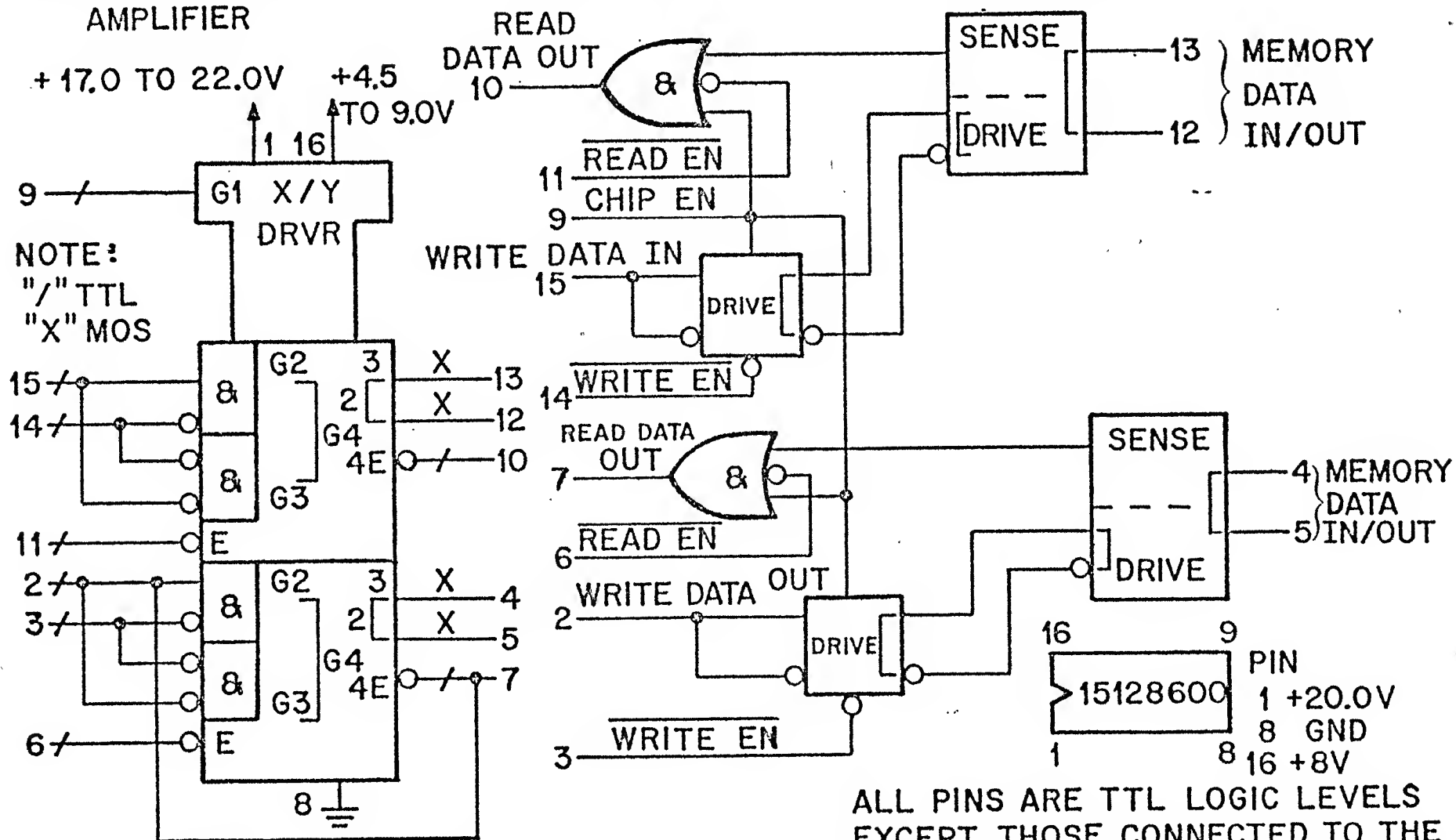
# 2X2 MOS MEMORY



# TTL LOGIC

SN 75370  
BIT DRIVER/SENSE  
AMPLIFIER

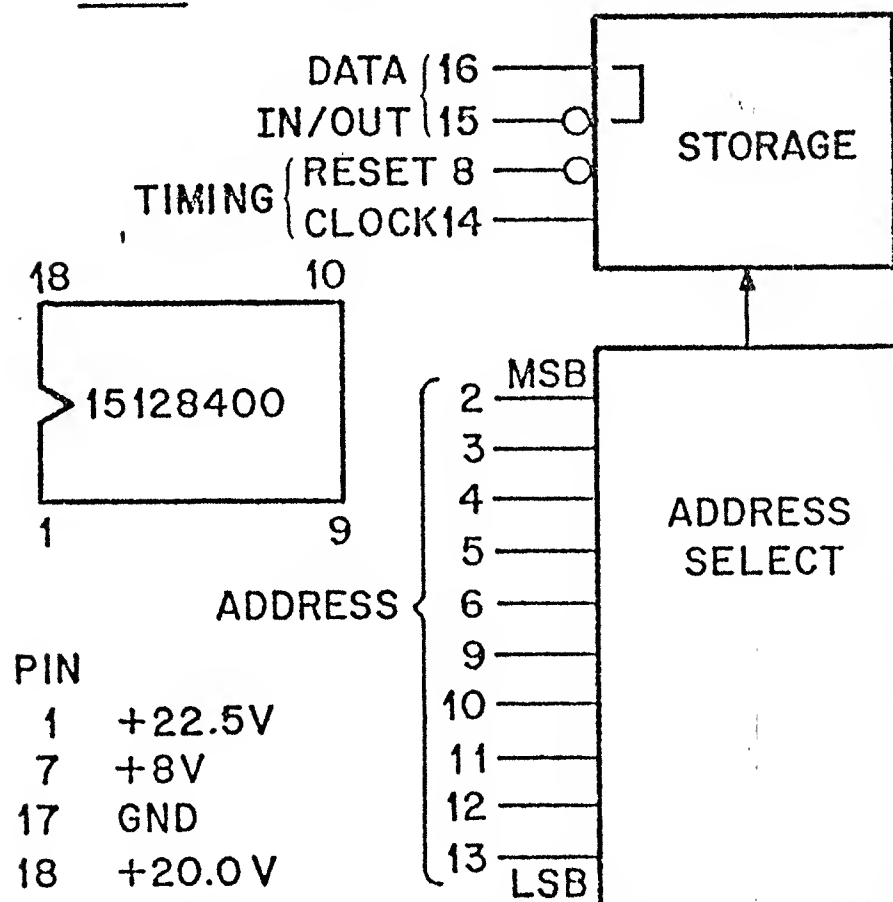
75370 DUAL BIT DRIVER / SENSE AMP



ALL PINS ARE TTL LOGIC LEVELS  
EXCEPT THOSE CONNECTED TO THE  
MEMORY CHIP.

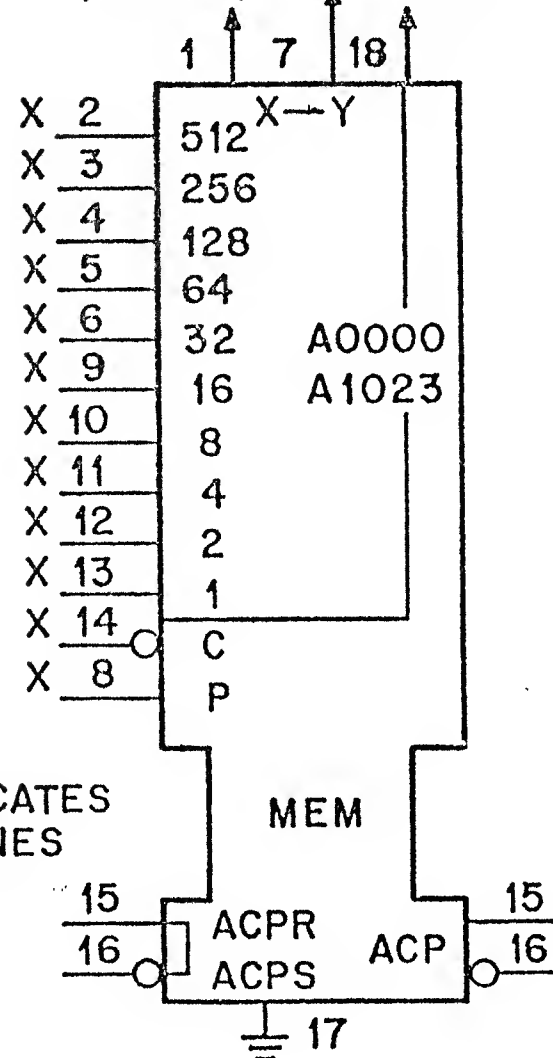
# MOS CHIP HANDOUT

6002 1024 X 1 BIT MOS MEMORY

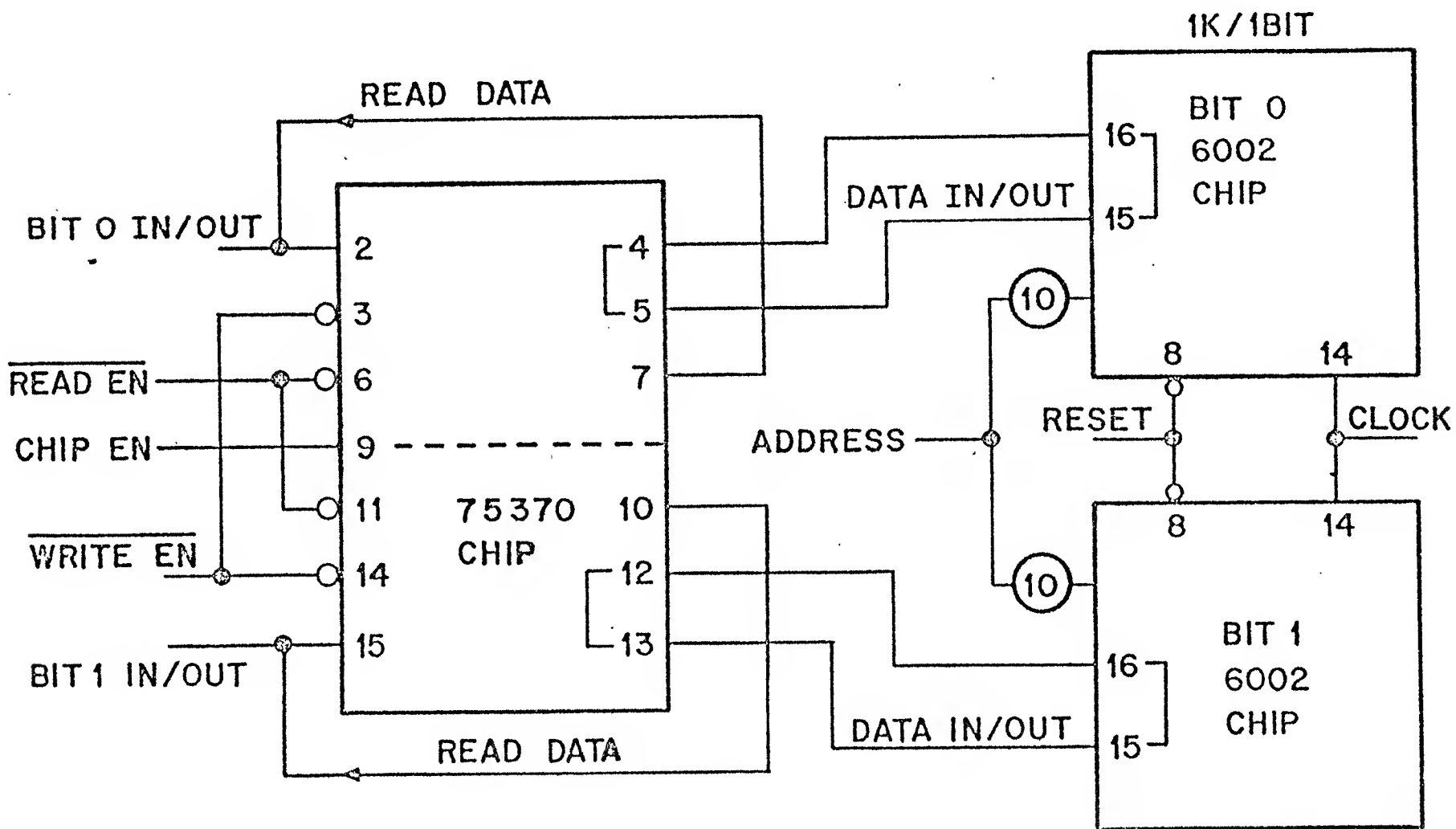


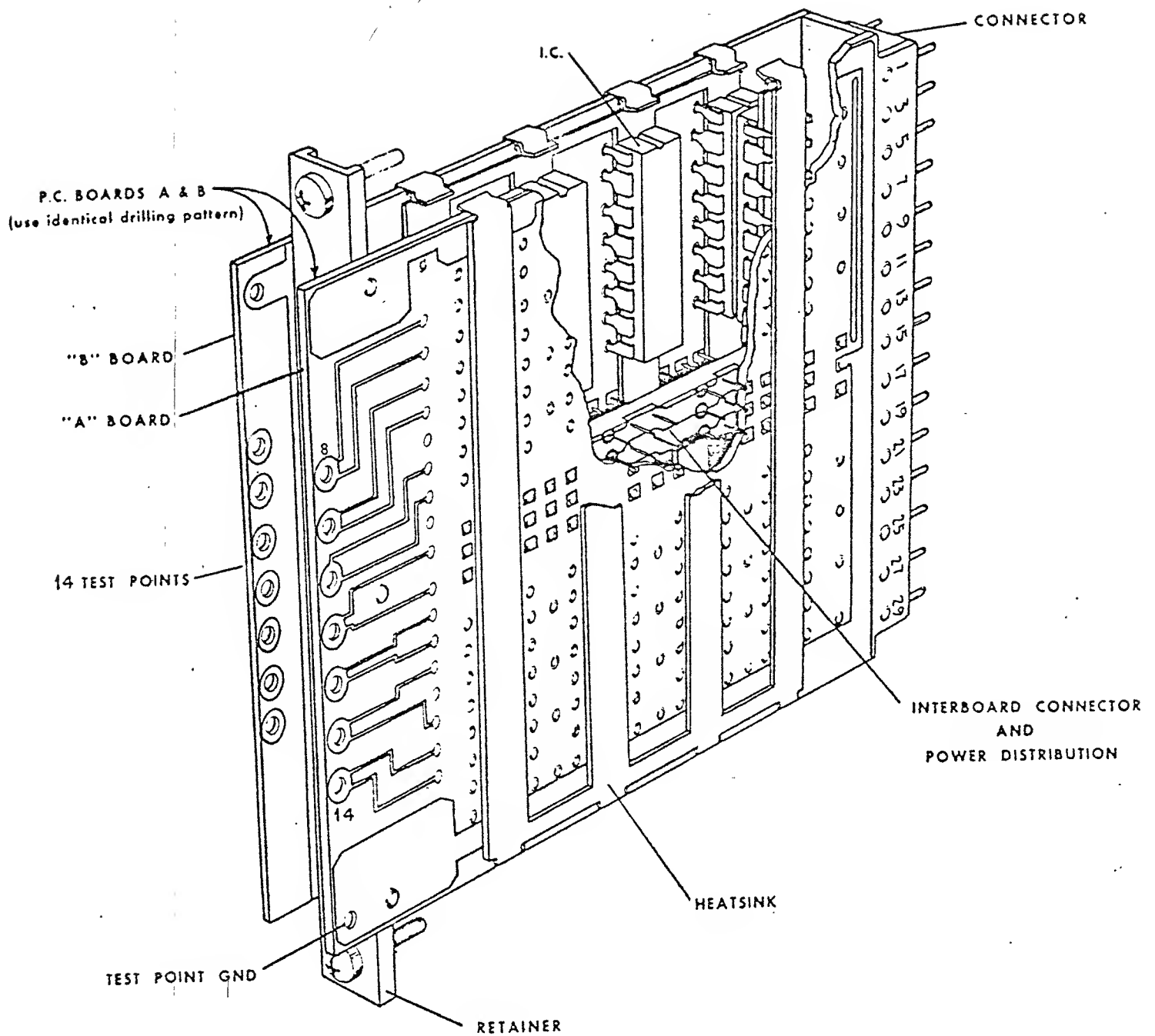
ADDRESS AND TIMING INPUTS  
ARE MOS LOGIC LEVELS.

6002 1024 WORD X 1 BIT  
MOS MENOS MEMORY  
(+22.5V) (+8.V) (+20.0V)



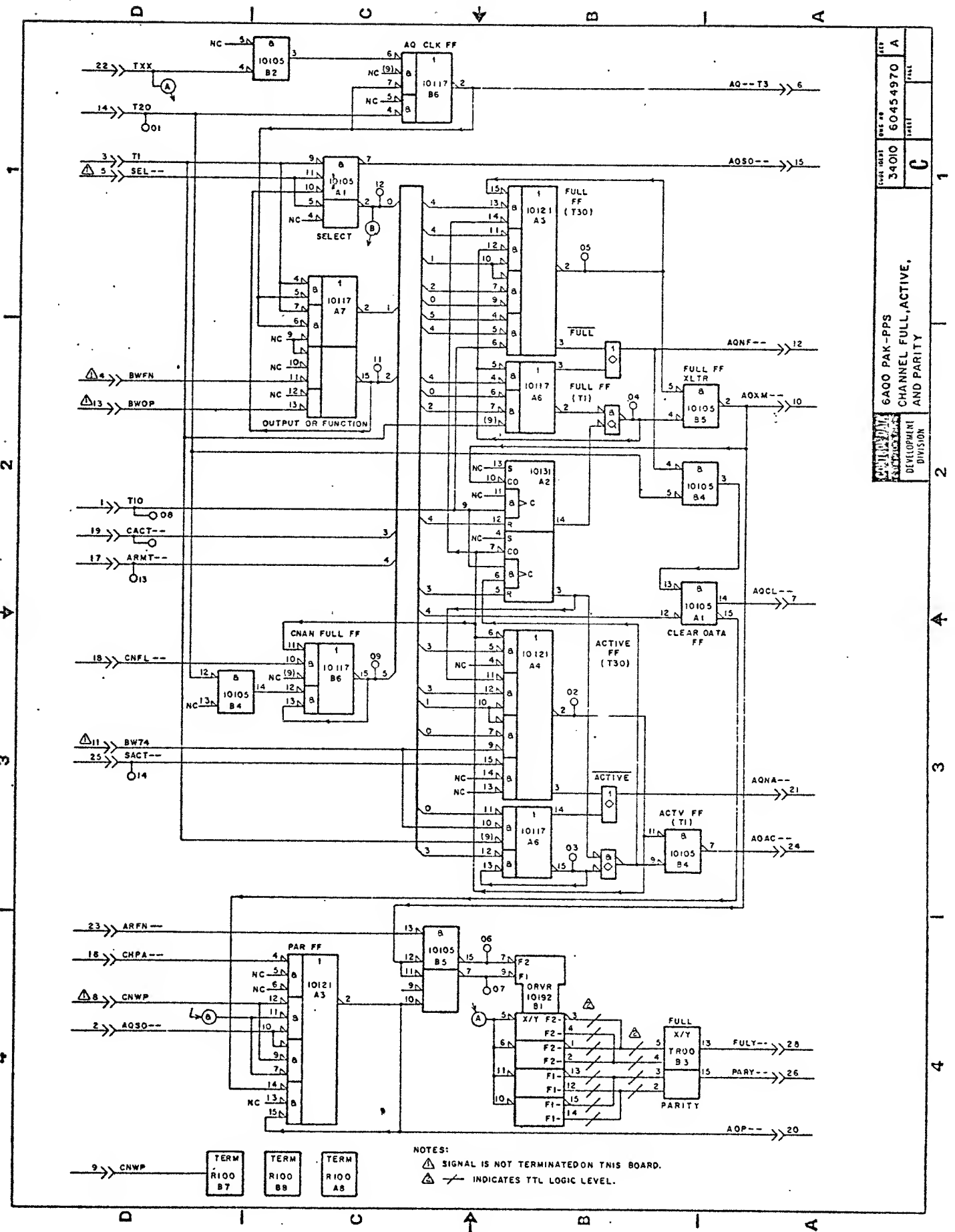
# TYPICAL MOS MEMORY CIRCUIT





## ECL CYBER170 MODULE - 16 PAK

30 PIN CONNECTOR 14 TEST POINTS, 16 I.C.'S,



## CABLETAD REPORT. C1J15, JACK.

ADD/DELETE	PANEL	NET	COLORS	LENGTH	CBL END	TAO END
------------	-------	-----	--------	--------	---------	---------

	C1	\$DCP000		024	C1J15-A01/A02	01M39-12/M10
	C1	\$DCP001		024	C1J15-A03/A04	01M39-04/M02
	C1	\$DCP002		024	C1J15-A05/A06	01M39-16/M14
	C1	\$DCP003		024	C1J15-B01/B02	01M39-08/M06
	C1	\$DCP004		024	C1J15-B03/B04	01M39-15/M13
				024	C1J15-B05/B06	01M39-11/M09
					C1J15-C01/C02	01M39-07/M05
						01M39-03/M01

## NET LIST REPORT.

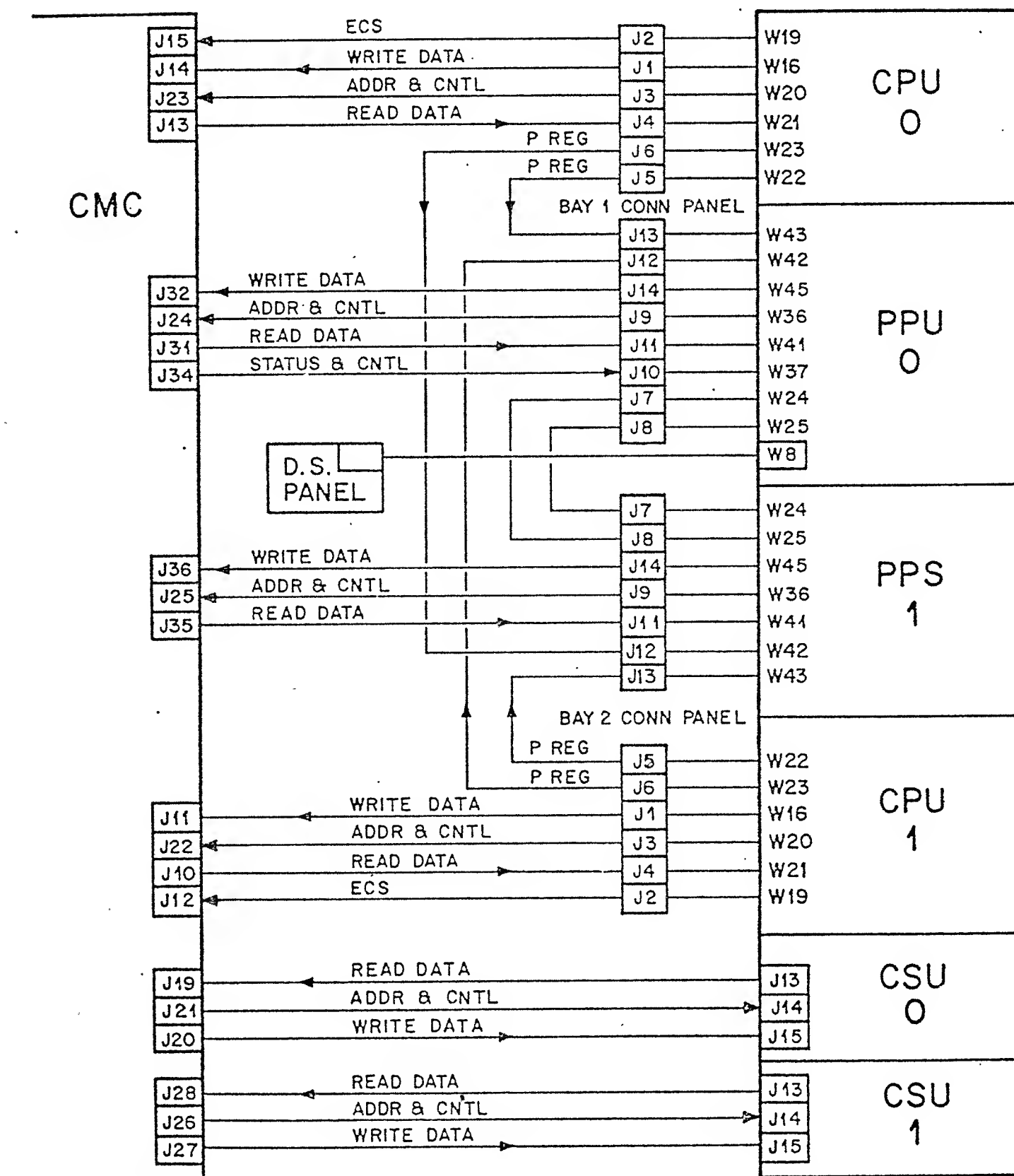
NET	LENGTH	CBL	ORIGIN	DESTINATION	COLORS	ADD/DELETE
-----	--------	-----	--------	-------------	--------	------------

AA0705						
AB17	7					
AB1701	7		01D09-12/			
AB1702	5		01E29-01/	01C05-22/M		
AB1703	5		01B31-11/	01F27-24/M		
AB1704	5		01032-11/	01C35-02/M		
AB1705	5		01B33-11/	01C35-04/M		
			01B34-11/	01C35-08/M		

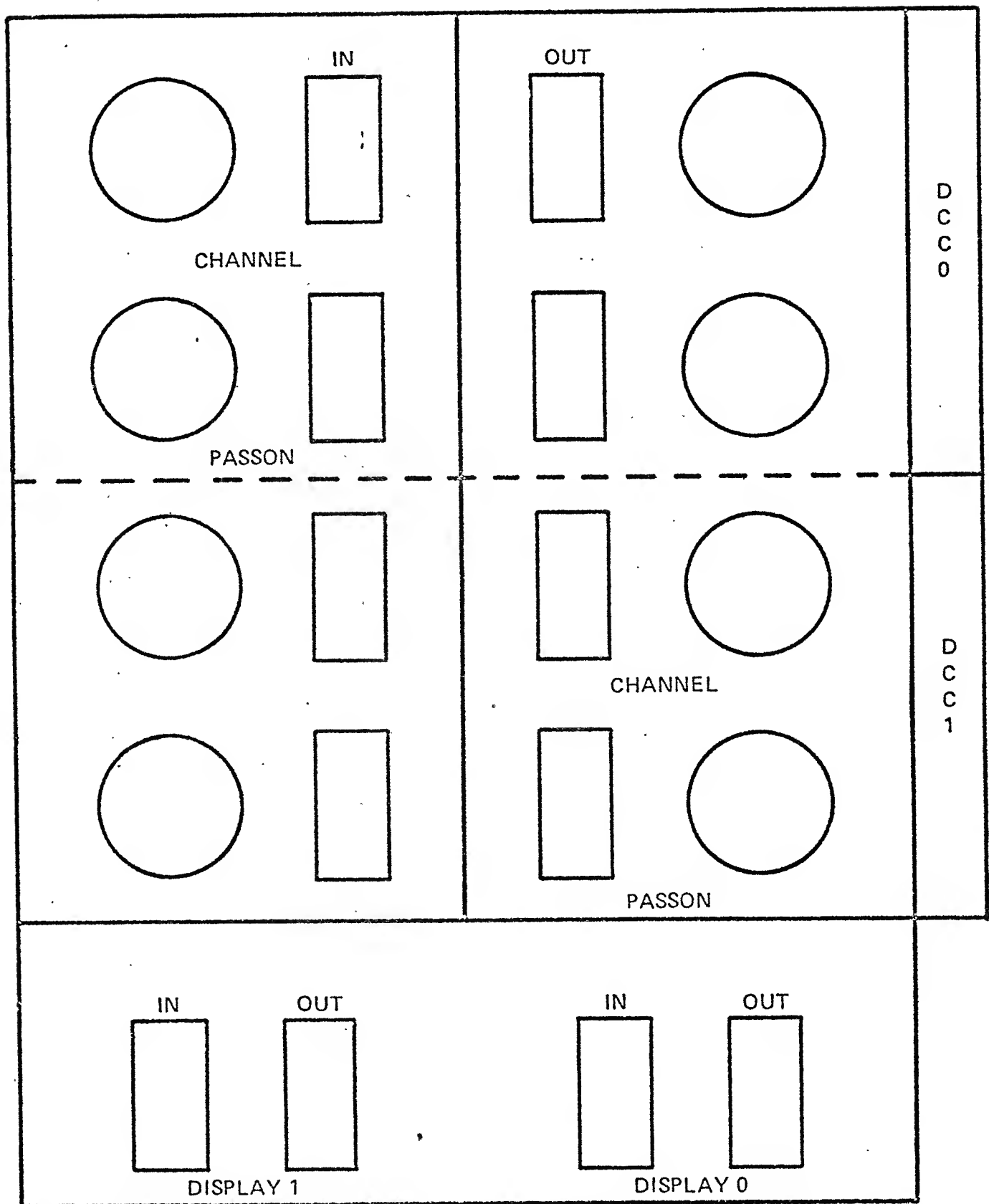
## CROSSTAB REPORT. PACK 01P40.

\$ACPCPD	24	C1	01P40-02/M04	C1J14-K01/K02
\$ACPCPD	24	C1	01P40-02/M04	C1J14-K01/K02
\$BPECPO	24	C1	01P40-08/M06	C1J14-J05/J06
\$BPECPO	24	C1	01P40-08/M06	C1J14-J05/J06
\$LPEOXD	24	C1	01P40-11/M09	C1J14-M03/M03
\$LPEOXD	24	C1	01P40-11/M09	C1J14-M03/M03
\$KMFO	24	C1	01P40-15/M13	C1J14-M04/M04
\$PEPCPO	24	C1	01P40-16/M14	C1J14-J03/J04
\$KMFO	24	C1	01P40-15/M13	C1J14-M04/M04
\$PEPCPO	24	C1	01P40-16/M14	C1J14-J03/J04
NMASCL	17		01P40-17/M	01L40-17/
ACCEPT	5		01P40-18/M	01Q38-17/
IOTPAR	9		01P40-19/M	01N39-10/
BKPTFL	27		01P40-20/M	01I35-19/
REOEXJ	41		01P40-21/M	01F27-03/
QATROY	5		01P40-22/M	01Q37-17/
MONFLG	39		01P40-23/M	01F27-26/
PAPERR	49		01P40-24/M	01C26-24/
TLKRO9	19		01P40-26/M	01L41-26/M

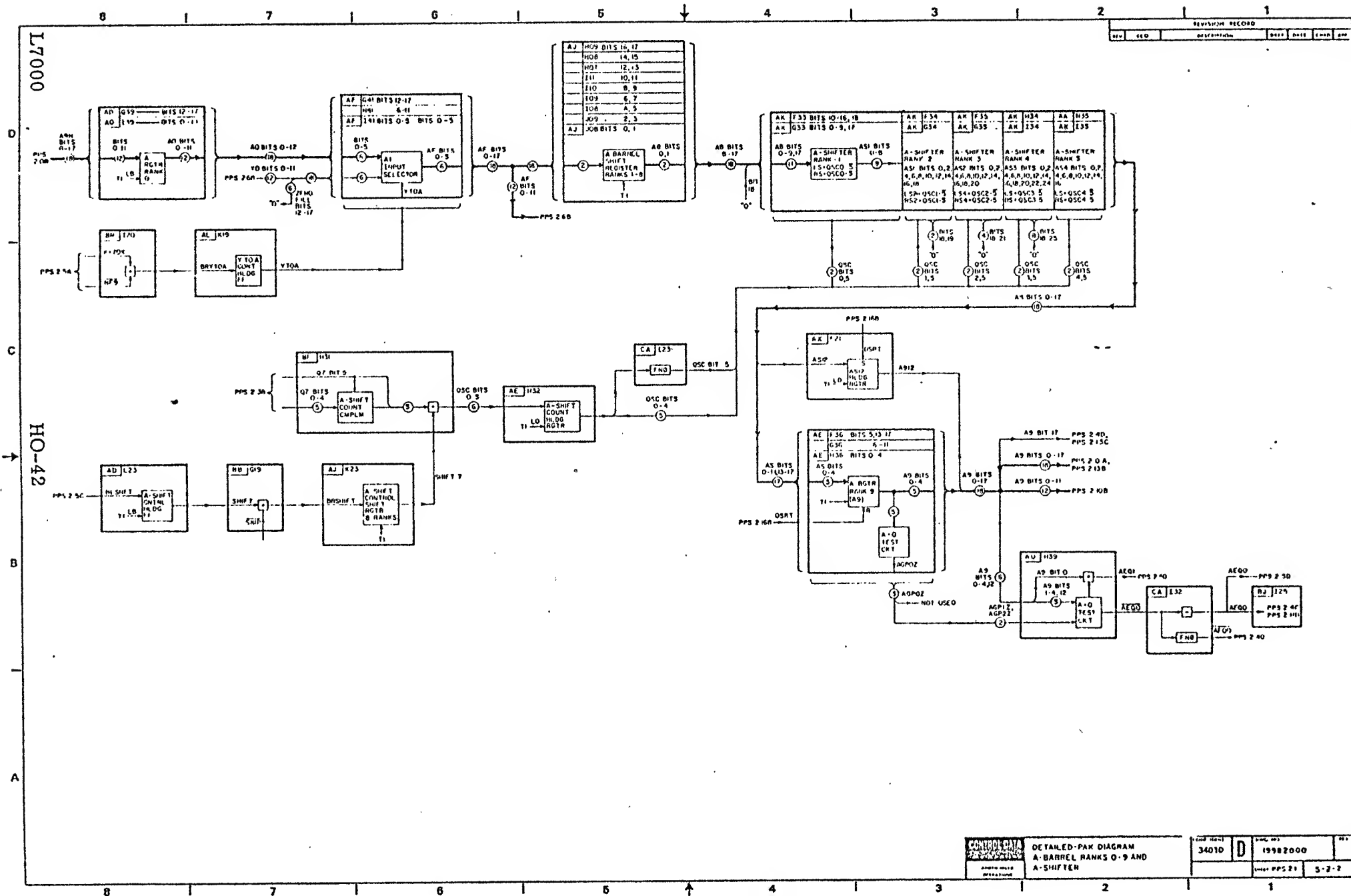
# CYBER 172, 3, 4 CABLE DIAGRAM



# BOTTOM PORTION OF THE CHANNEL CONNECTOR PANEL



## DETAILED PACK DIAGRAM



## PRIMARY BLOCK DIAGRAM MASTER CLOCK

The master clock contains a 10-MHz crystal oscillator that generates the internal reference frequency to run synchronously all components of the computer system. If the computer system has the ECS standard option installed, the ECS controller provides the 10-MHz ECS master clock 25 nanosecond pulses (external reference frequency) to the master clock. This synchronizes all components of the computer system with all ECS banks. In the event the ECS master clock fails, the reference frequency is generated internally by the master clock.

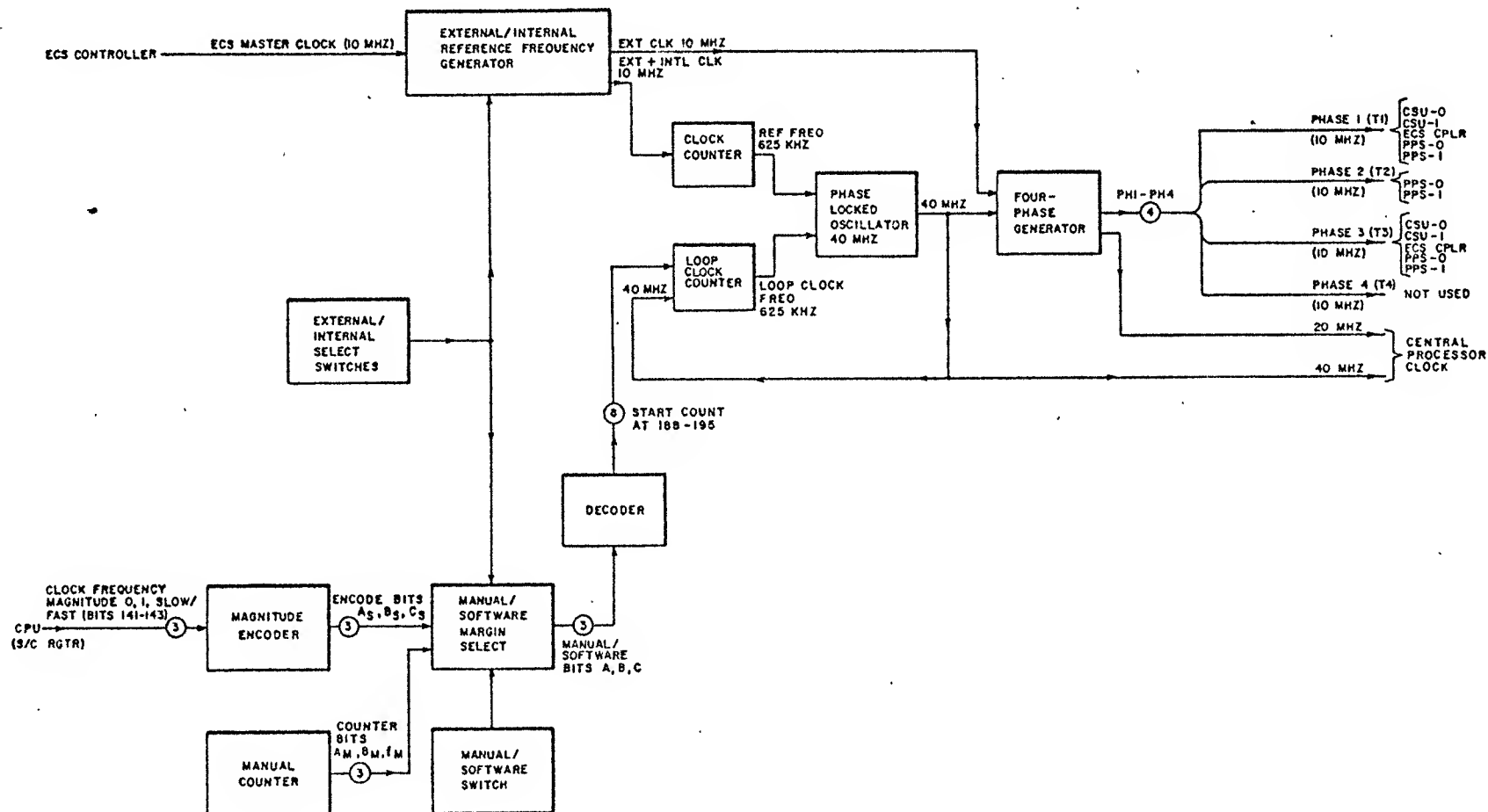
The master clock, using either the external or internal reference frequency as its source, provides 10-MHz pulses to the clock counter. The reference frequency is divided by 16 in the clock counter which then outputs 625 kHz with a time period of 1.6 microseconds to the phase-locked oscillator. For a normal frequency output of 40 MHz from the phase-locked oscillator, the loop clock counter must be set at 64. The loop clock counter output, therefore, is 40 MHz divided by 64 which is 625 kHz with a time period of 1.6 microseconds. The phase-locked oscillator compares the leading edge of these two inputs and generates a dc voltage based on the time difference. This dc voltage controls a voltage-controlled oscillator in the phase-locked oscillator circuit. If the 40-MHz output frequency changes, the resultant output change of the loop clock counter is detected as a time difference between the leading edges of the two pulses by the phase-locked oscillator. The difference causes a change in dc voltage that affects the voltage-controlled oscillator and changes the

frequency back to the normal output. The 40 MHz is sent to the four-phase generator which outputs 25-nanosecond pulses at a 10-MHz rate in four phases (time frame). The four-phase generator also outputs a 50-nanosecond square wave at a 20-MHz rate. The 10-MHz output is fanned out as an ac signal to the various components of the computer system.

The external/internal select circuit is a switching circuit that allows selecting the internal or external generation of the reference frequency. The switch is in the internal position if the computer system does not have the ECS standard option installed or when frequency margins are performed. Frequency margins are controlled by software via the status and control register (clock frequency magnitude bits) or manually generated by switches on the master clock packs. Switches for manual operation of the frequency margin checks are part of the margin select circuit.

The master clock receives a master clear signal from CMC to clear internal circuits of the master clock for synchronization. A master clear signal is also routed to the ECS coupler.

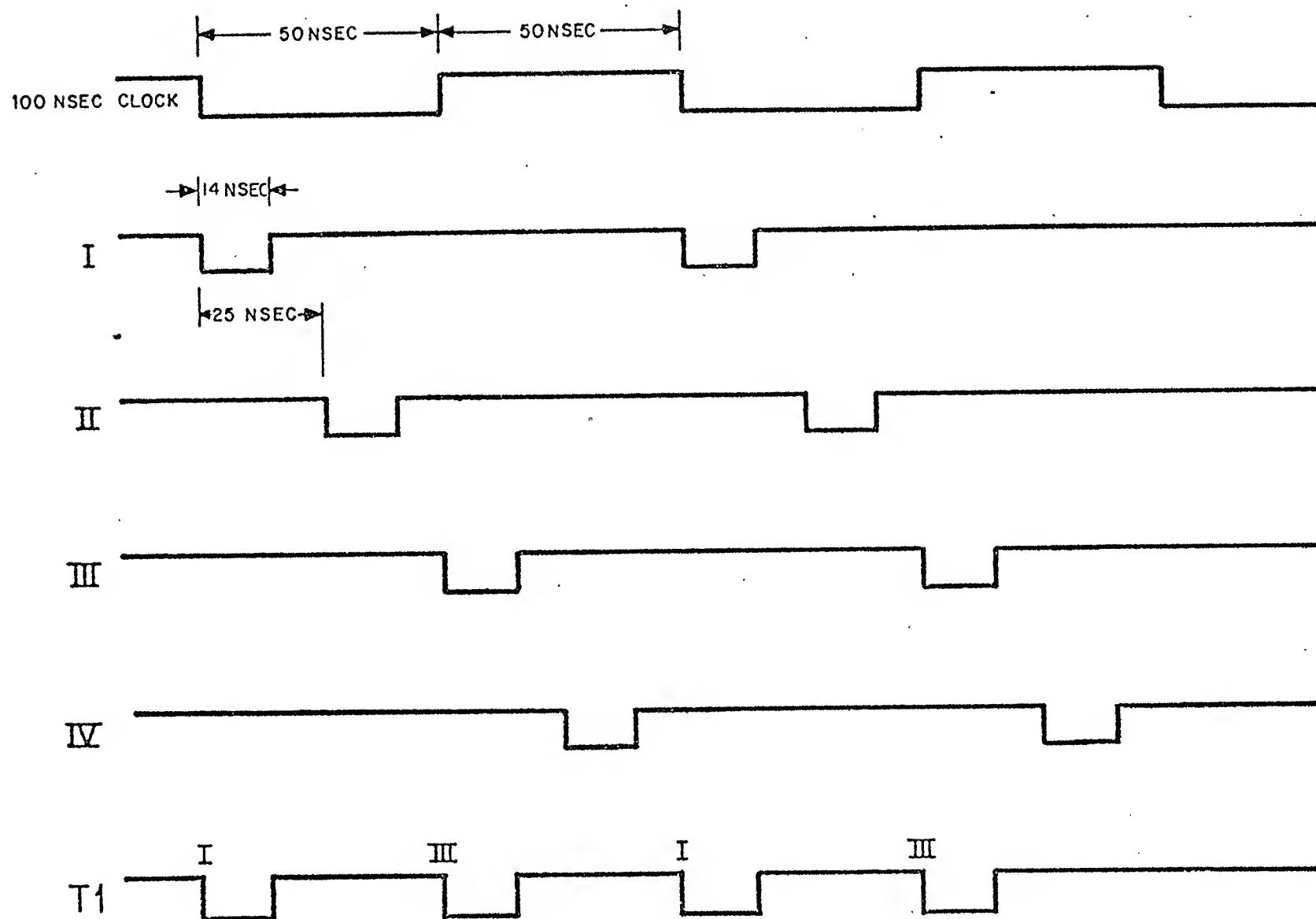
The master clock contains a +5.0-volt dc power supply to provide the TTL logic in the phase-locked oscillator with power. The standard -5.2 volts dc is converted to ac, then rectified and filtered for the +5.0 volts dc.



# CLOCK REQUIREMENTS FOR THE CYBER 172, 173, 174

CHASSIS	10 MHZ CLOCK				
	PHASE 1	PHASE 2	PHASE 3	PHASE 4	TOTAL
CPU-0	3		3		6
CPU-1	3		3		6
CMC	2		2		4
ECS COUPLER	1	1	1	1	4
PPS-0	2	1	2		5
PPS-1	2	1	2		5
CSU-0	1		1		2
CSU-1	1		1		2
TOTAL:	15	3	15	1	34

## CYBER 170 CLOCK



# DETAILED PAK DIAGRAM (PPS 3.20)

## PPS CLOCK DISTRIBUTION

### DESCRIPTION

The circuitry shown in PPS 3.20 generates all clocks used in the PPS, DCCs and display controller. Three clock phases from CMC (PH1, PH2, PH3) are converted to ECL levels, with 25-ns on times. The AA paks shorten the pulse widths of these clocks and fan out the resultant signals to the complete PPS chassis. Clock timing is illustrated in figure 4-2-13.

Shown in table 4-2-4 are the clock names, the signals from which they were developed their periods and on times.

Note that the clock names shown in this table do not necessarily apply to DSC/DCC clocks. As shown on the top three paks of the AA stack in PPS 3-20, T10 is sometimes generated by TB20, and T20 may be generated by a delayed TB20 for the DSC and DCC.

In addition, several specialized clocks are generated in the DC paks (R38, R39). These are used in DCC/DSC pass-on and pass-back timing.

The AC modules use T20 and MJE to generate a 10-MHz clock (10-MHz 00-13) and a 1-MHz clock (1-MHz 00-13) for each channel.

Timing for these channel clocks is shown in figure 4-2-14.

Clocks received from CMC have a 100-ns period. The 50-ns period of TB50, reproduced in T1, is generated by ORing PH1 and PH3.

TABLE 4-2-4. PPS CLOCKS

CLOCK	PERIOD	ON-TIME	DEVELOPED FROM
T1	50 ns	$12 \pm 2$ ns	TB50 ← PH1 + PH3
T10	100 ns	$12 \pm 2$ ns	TB10 ← PH1
Txx	100 ns	$\approx 25$ ns	TB30 ← PH3
T20	100 ns	$12 \pm 2$ ns	TB20 ← PH2
1 MHz	1 $\mu$ s	25 ns	MJE
10 MHz	100 ns	25 ns	TB20 ← PH2

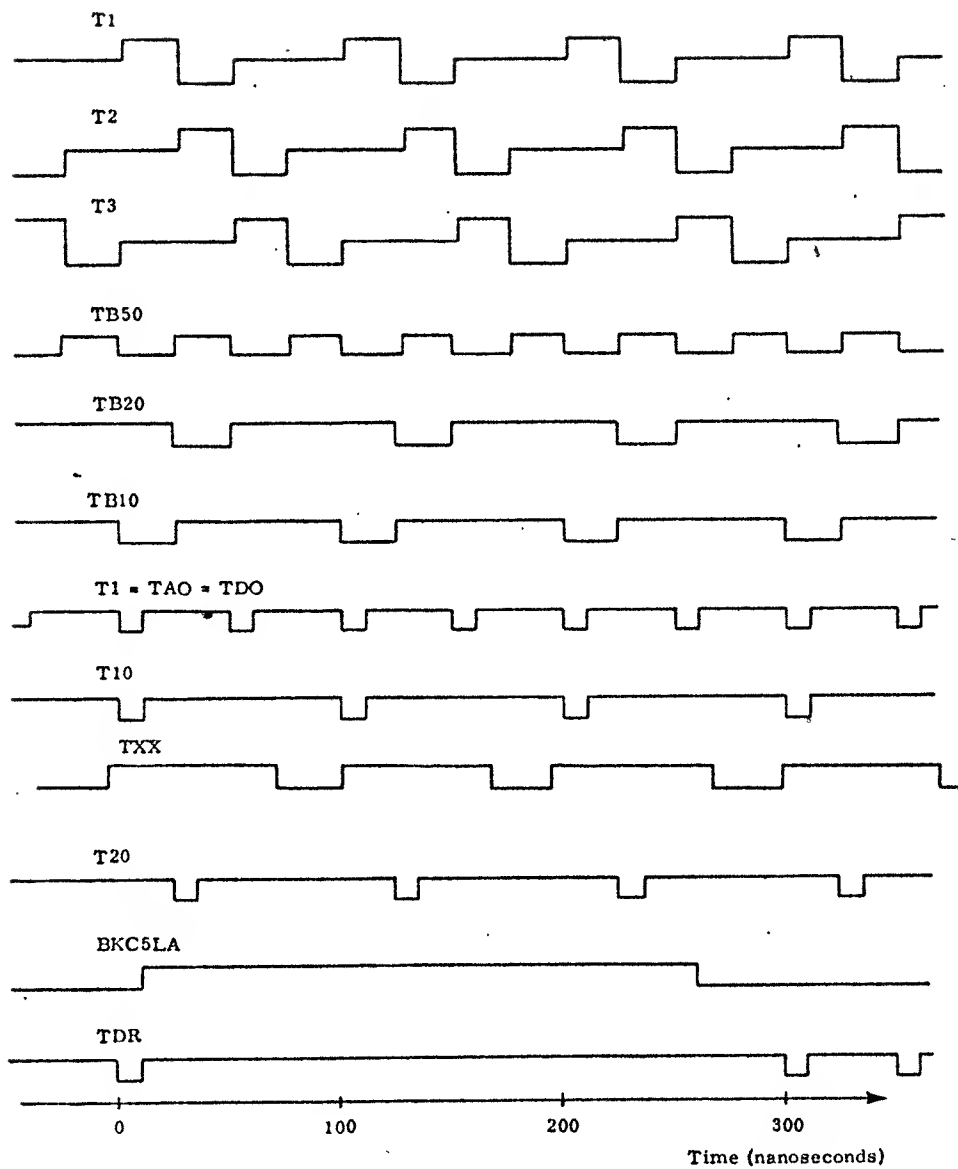


Figure 4-2-13. Basic PPS Clocks

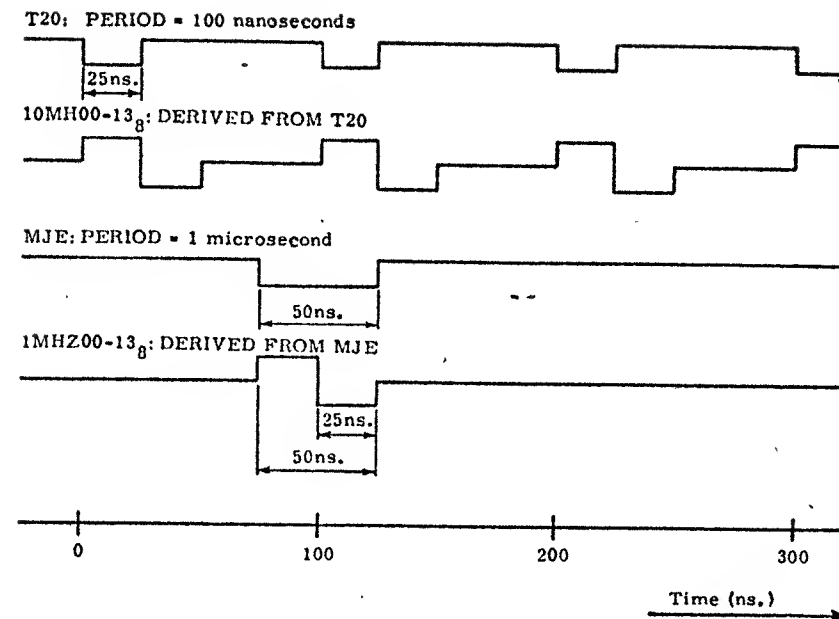


Figure 4-2-14. Channel Clocks

# INTRODUCTION

## SYSTEM BLOCK DIAGRAM

The system block diagram shows the relationship of the peripheral processor subsystem (PPS) to other functional units within the CDC CYBER 170 Computer Systems. This diagram also relates physical characteristics to functional characteristics at the chassis level and defines the boundaries of the publications that form the CDC CYBER 170 hardware maintenance manuals.

Each PPS interfaces with:

- The other PPS
- Central memory control (CMC)
- 12 data channels
- Both central processors (CPU-0 and CPU-1)
- Extended core storage (ECS)

Any peripheral processor (PP) may access any channel in either PPS, and all channels may be active at once.

Channel 10<sub>8</sub> is dedicated to a display controller (DSC). Data channel converters (DCCs) connect to two arbitrarily selected channels in the PPSs.

One channel in each PPS interfaces with the distributive data path (DDP). The DDP allows the PPs of any CDC CYBER 170 to access ECS.

Channel 14<sub>8</sub> is a real-time clock that may be read by any PP.

Channel 16<sub>8</sub> is the status and control register (SCR). The SCR is a collection of bits where any one bit may indicate the status of a particular piece of equipment or control the mode of operation of a central computer function.

SCR-0 bits set either through force set signals from various sources in the computer or by channel instructions. They affect functions throughout the computer.

SCR-1 bits pertain only to PPS-1; SCR-1 therefore controls internal PPS-1 functions only.

Each PPS sends data and controls to the other PPS for channel input/output (I/O) operations. PPS-0 also transmits certain SCR control bits and dead-start information to PPS-1.

CMC sets and resets certain SCR bits. It also transmits and receives data and control signals for data interchange with both PPS-0 and PPS-1.

Both CPUs send the P register contents to PPS-0 and PPS-1. They also send a run status bit to SCR-0. PPS-0, in turn, sends control bits to the CPUs from both the SCR and other time-shared hardware.

Internally, each PP comprises a number of parallel registers (A, Q, P, and K are the major ones) at any given time. The data representing any particular PP shifts circularly at 50-nanosecond intervals into the registers formerly occupied by the next lower-numbered PP. Each PP is assigned to a peripheral processor memory (PPM) at deadstart time and accesses that PPM only.

At one of the 10 stages of the circular shift register, the PP accesses time-shared hardware that performs arithmetic, logical I/O, and control operations on the PP's registers.

This situation is depicted in figure 5-1-1 by a rotating wheel, with each segment of the wheel representing a PP. At slot time, a PP rotates into position to use the time-shared hardware. The contact brush shown denotes the interaction between PP and shared hardware.

## PPS-0 AND PPS-1 DIFFERENCES

PPS-0 is similar to PPS-1. However, certain standard features in PPS-0 are optional in PPS-1. PPS-0 has 2 DCCs and 10 PPs, whereas PPS-1 may have 0 to 2 DCCs and 4, 7, or 10 PPs.

PPS-1 does not have, optionally or otherwise, a display controller, a deadstart panel, or a real-time clock. Additionally, PPS-1 contains only a smaller version of the SCR found in PPS-0.

Similarities between the two PPSs are shown in the system block diagram (figure 5-1-2). Differences, comprised mainly of those things PPS-1 lacks relative to PPS-0, are shown in the simplified system block diagram (figure 5-1-1).

The various PPS-1 configurations are achieved by omitting unneeded memory, DSC, DCC, and deadstart panel interface packs. This, combined with a relatively small number of wiring modifications to the PPS-0 design, is the physical difference between PPS-0 and PPS-1.

The PPS-0 and PPS-1 relationship is nearly symmetrical. In most cases, signals from one chassis have counterpart signals originating in the other chassis from identical hardware. This means that for most inter-PPS signals, a corresponding signal with reciprocal origins and destinations exists. Symmetry extends to internal hardware which is predominantly identical in both chassis.

The minor nonsymmetries between the PPSs are:

- PPS-1 lacks several functional parts contained in PPS-0.
- The PPS-1 channels must be distinguishable from PPS-0 channels for software purposes.
- Any one of the PPMs in either PPS must be capable of reading the deadstart panel. Four switches on the panel determine which PPM performs this function.
- The deadstart signals and the PP counters in each PPS are synchronized, with PPS-0 and PPS-1 being master and slave, respectively.

## MULTILEVEL DIAGRAMS

The diagrams for each functional unit in the PPS are provided at four levels: primary block diagram (PBD), secondary block diagram (SBD), detailed pak diagram (DPD), and logic diagram (LD).

The PBD shows a high-level relationship between the various functional components (register, adders, and so on).

The SBD is a functional-to-physical bridge that shows each pak and all pak-interconnecting data and control signals.

The DPD is a functional-to-physical bridge that shows:

- Functional detail of every unique pak
- Data paths within the pak
- All pak-interconnecting data and control signals
- Backup text for each diagram
- A chart containing test points for all major signal paths on the diagram

Interchassis signals are shown on the PBDs, SBDs, and DPDs. Unless otherwise stated, these signals may be assumed to have identical destinations, whether they originate from PPS-0 or PPS-1; the diagrams apply equally well to PPS-0 and PPS-1. In cases where the two systems differ, the basic drawing shows PPS-0; notes and dotted lines detail the PPS-1 configuration.

The LD depicts the unit at the integrated circuit (IC) level; it consists of one or more logic diagrams for each pak in the unit.

Each LD consists of IC identification and placement, IC interconnection, plus backpanel wiring information in the form of signal names for pak connector pins.

## PAK-TO-DIAGRAM CROSS-REFERENCE TABLES

These tables appear at the beginning of parts 2, 3, and 4; they list all PPS chassis pak types along with the following information on each.

Location	All chassis locations at which this pak type can be found for each PPS option
Diagram	DPDs on which this pak type is shown
Function	The function covered on the corresponding pak diagram

The tables enable a user to identify all functional uses of a particular pak type and show the availability and location of substitute paks during maintenance; they also help in locating all paks of the same type without having to scan the pak placement diagrams. Unless otherwise indicated, information provided in these tables applies to both PPS-0 and PPS-1.

## PAK PLACEMENT DIAGRAMS

These diagrams are included to aid in translating the PPS chassis into functional entities.

## KEY TO DIAGRAM SYMBOLS

The key to diagram symbols are shown in figure 5-1-7. The symbology and conventions were chosen to simplify or clarify; they are essential to the use and understanding of the diagrams. The AND and OR symbols define functions, not gates (for example, AND gates in the hardware may actually be shown as OR functions on the diagrams).



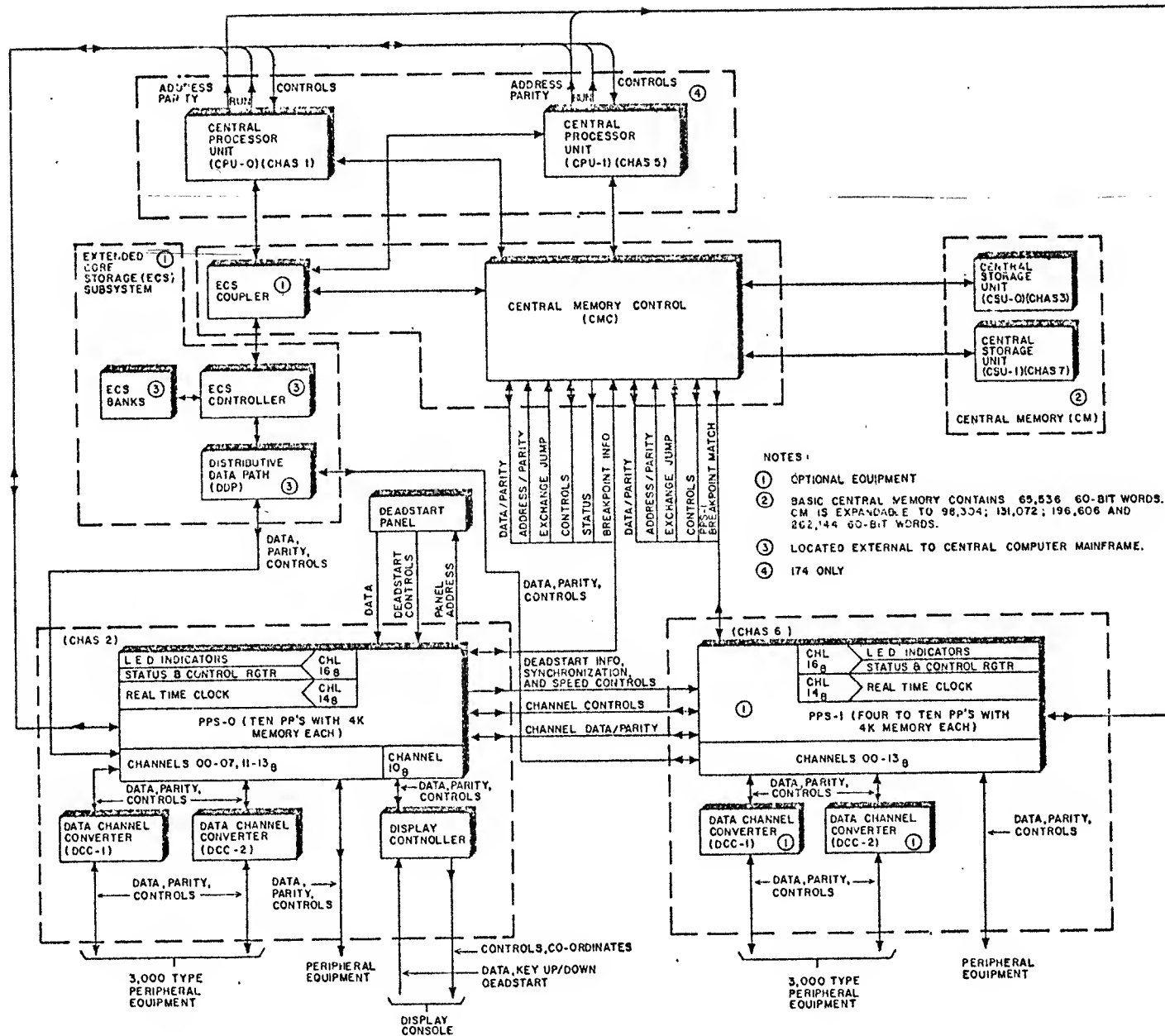
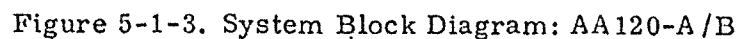


Figure 5-1-2. System Block Diagram: AA107-A, AA131-B

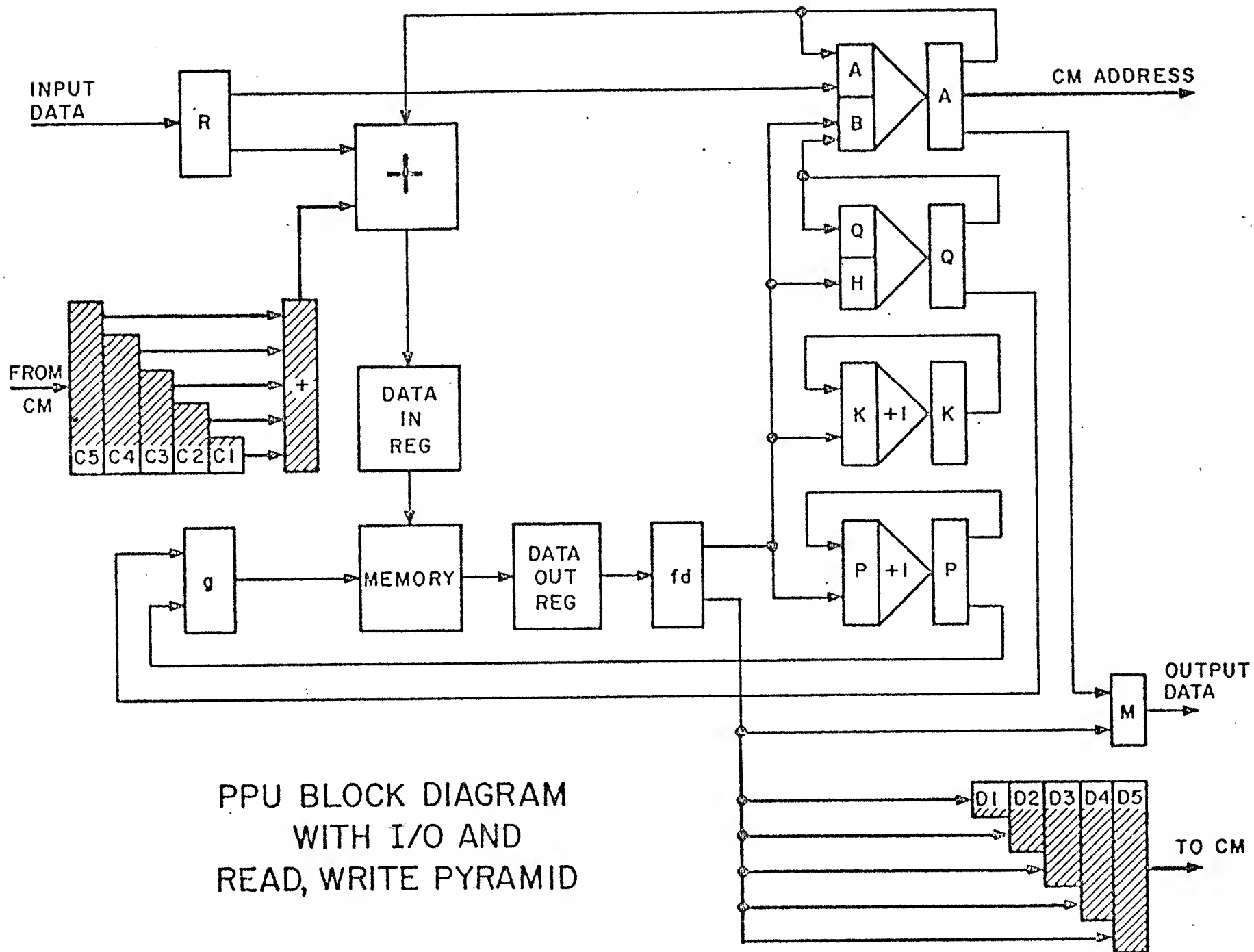


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
A																						
B				10B8	10U8	10U8		10B8	10U8	10U8		10B8	10U8	10U8		10B8	10U8	10U8		10B8	10U8	
C				10U8	10D8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8
D							10U8						10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8
E	4APD	4APD	TCED	TCED		TCED		3AN8	3AN8												3AN8	
F	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	7C7D	3X8D		4B8D	8K8D	7A8D			
G	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
H	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
I	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
J	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
K	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
L	3ACD	4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
M		4APD	4APD	4APD	8A8D	8A8D	4APD	4APD	8A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
N		4K8D	4K8D	4K8D	4K8D	4K8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
O	3K8D	4K8D	4K8D	4K8D	4K8D	4K8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
P	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D
Q	4E2D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D	3K8D
R	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8	30S8

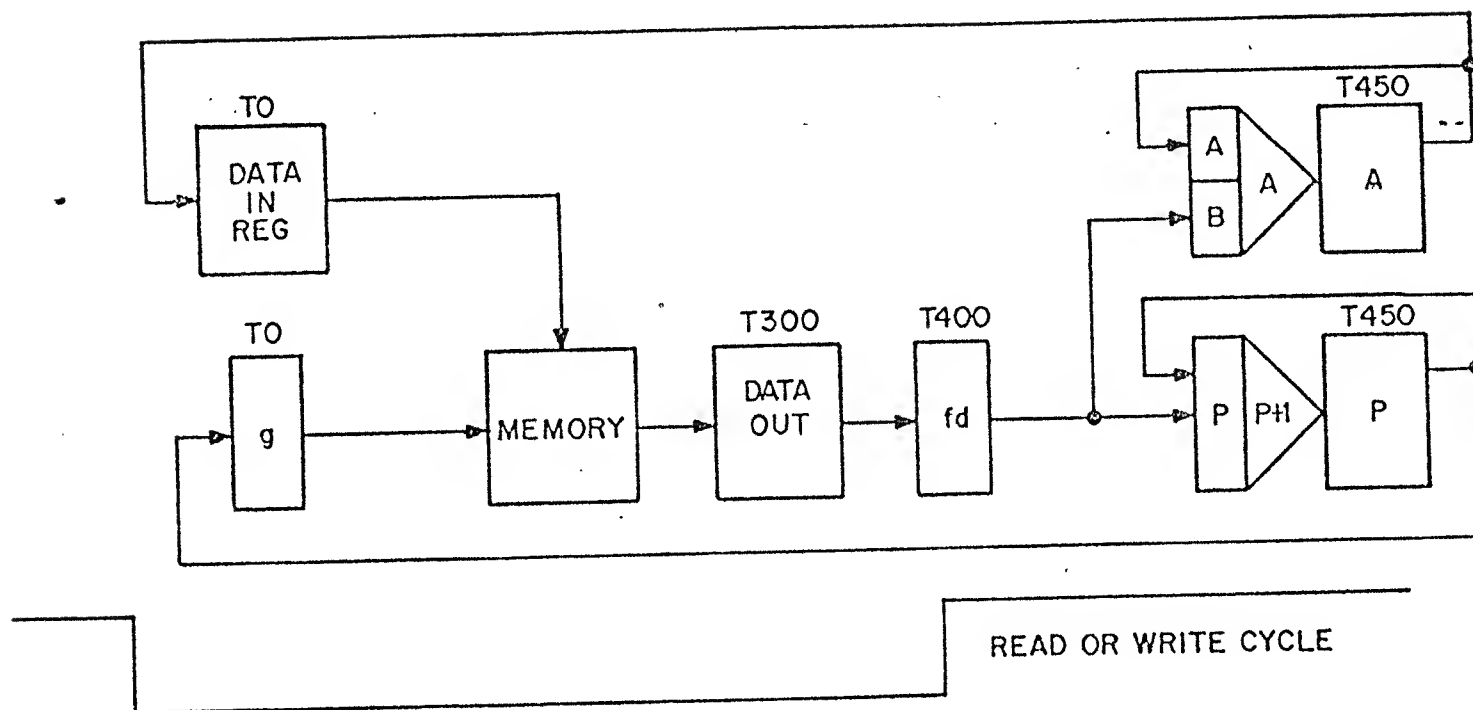
  

	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	
A																						
B	10U8		10U8	10U8		10U8	10U8		10U8	10U8		10U8	10U8	10U8		10U8	10U8	10U8		10U8	10U8	10U8
C	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8
D	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8	10U8
E																						
F	4B8D		4B8D	8A8D	7C7D	4B8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D	8A8D
G	3A8D	7A8D	3A8D	3A8D	4B8D	35L8	7A8D	4B8D		3A8D	3A8D	4A8D	4A8D	4A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
H	4A8D	4A8D	4A8D		4B8D	8A8D	3A8D	3A8D		4B8D	3A8D	4B8D	4A8D	4A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
I	3A8D	3A8D	8A8D	8A8D	3A8D	3A8D	3A8D	8A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	
J	3A8D	4B8D	8A8D	4B8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D				3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
K	4A8D	4A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
L	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
M	3A8D	3A8D		3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
N	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
O	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
P	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D
Q	4B8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D	3A8D
R	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D	4A8D

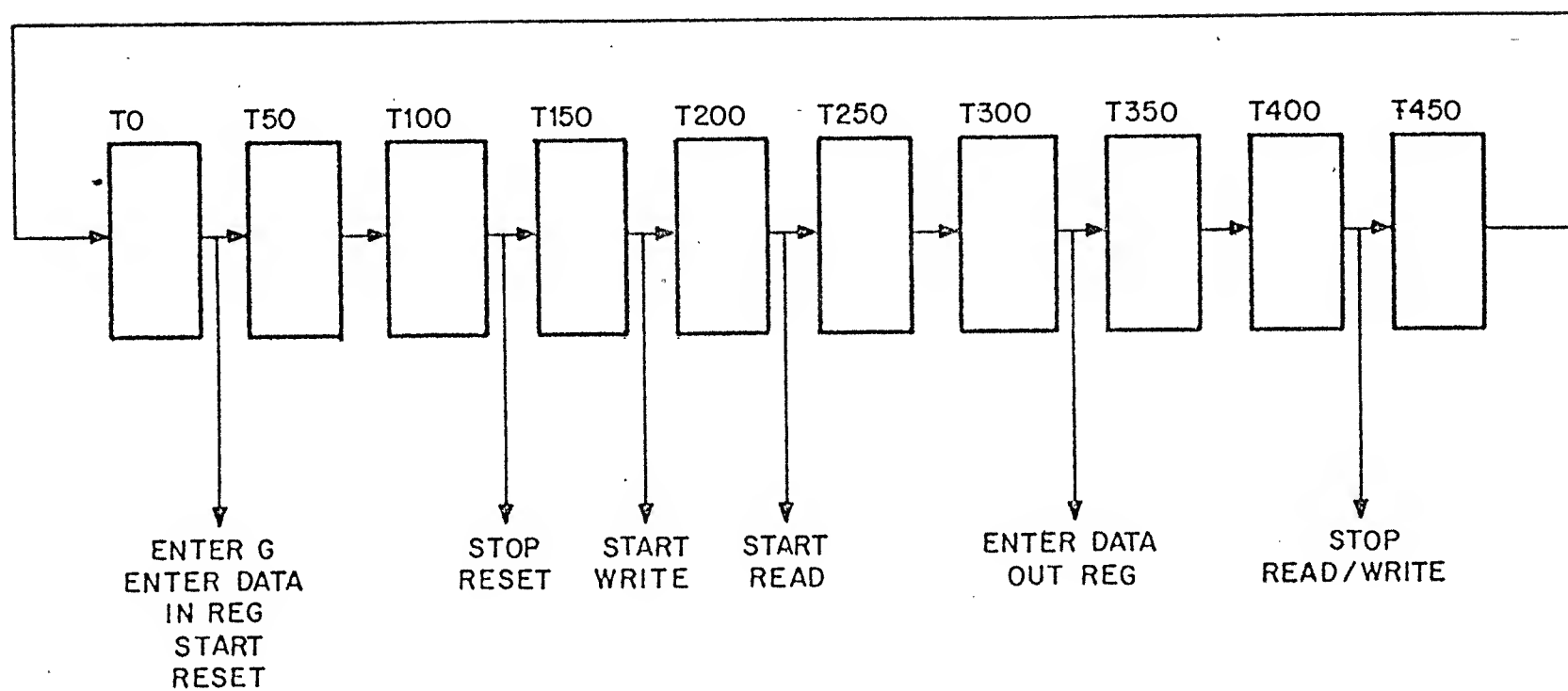
Figure 5-1-4. PPS-0 Pak Placement (Models 171 through 175, Chassis 2)



## PPU MEMORY REFERENCE

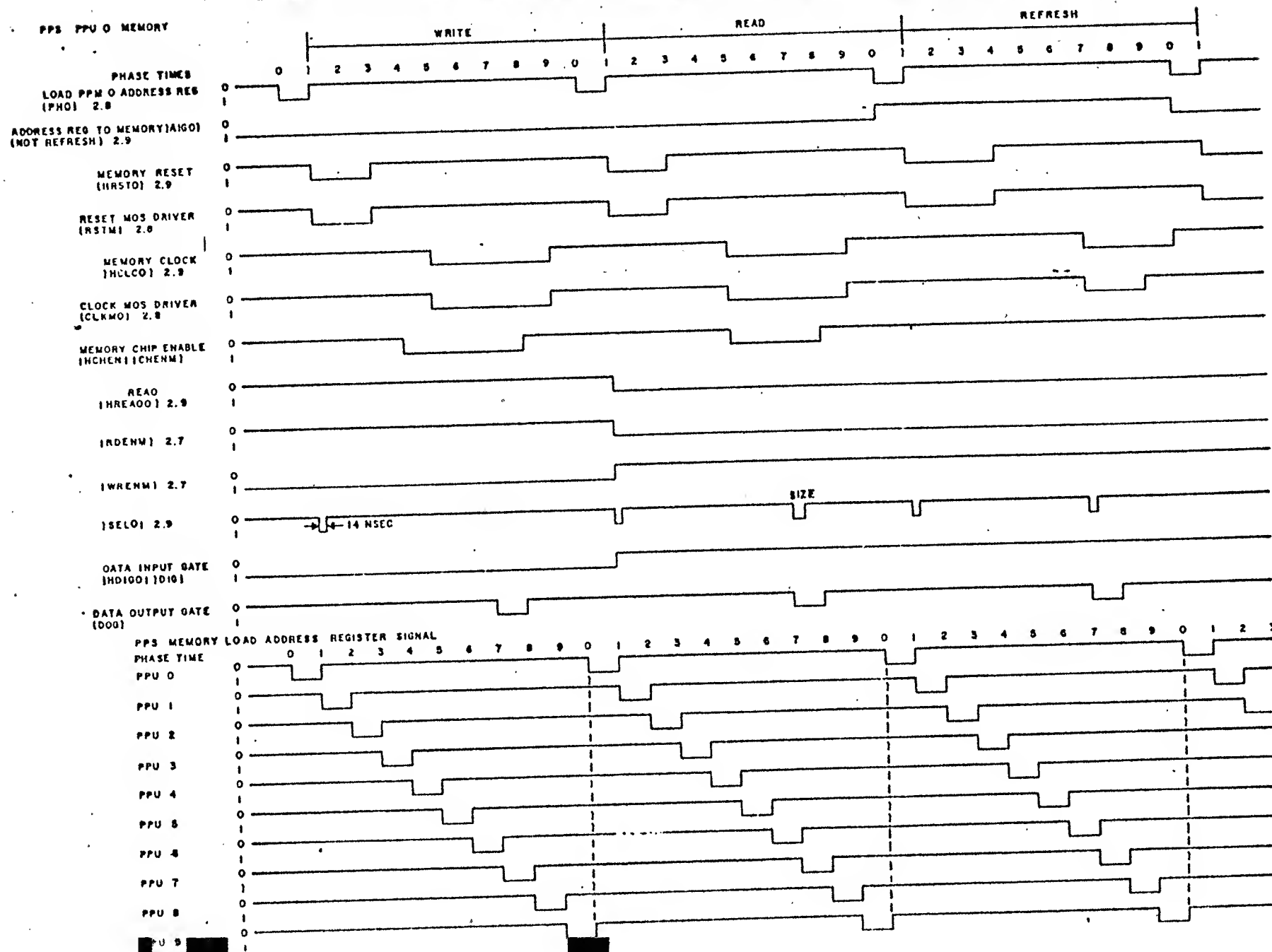


## PPU MEMORY TIMING CHAIN



# PPS MEMORY REFERENCE 2.5 A-1

PPS PPU 0 MEMORY



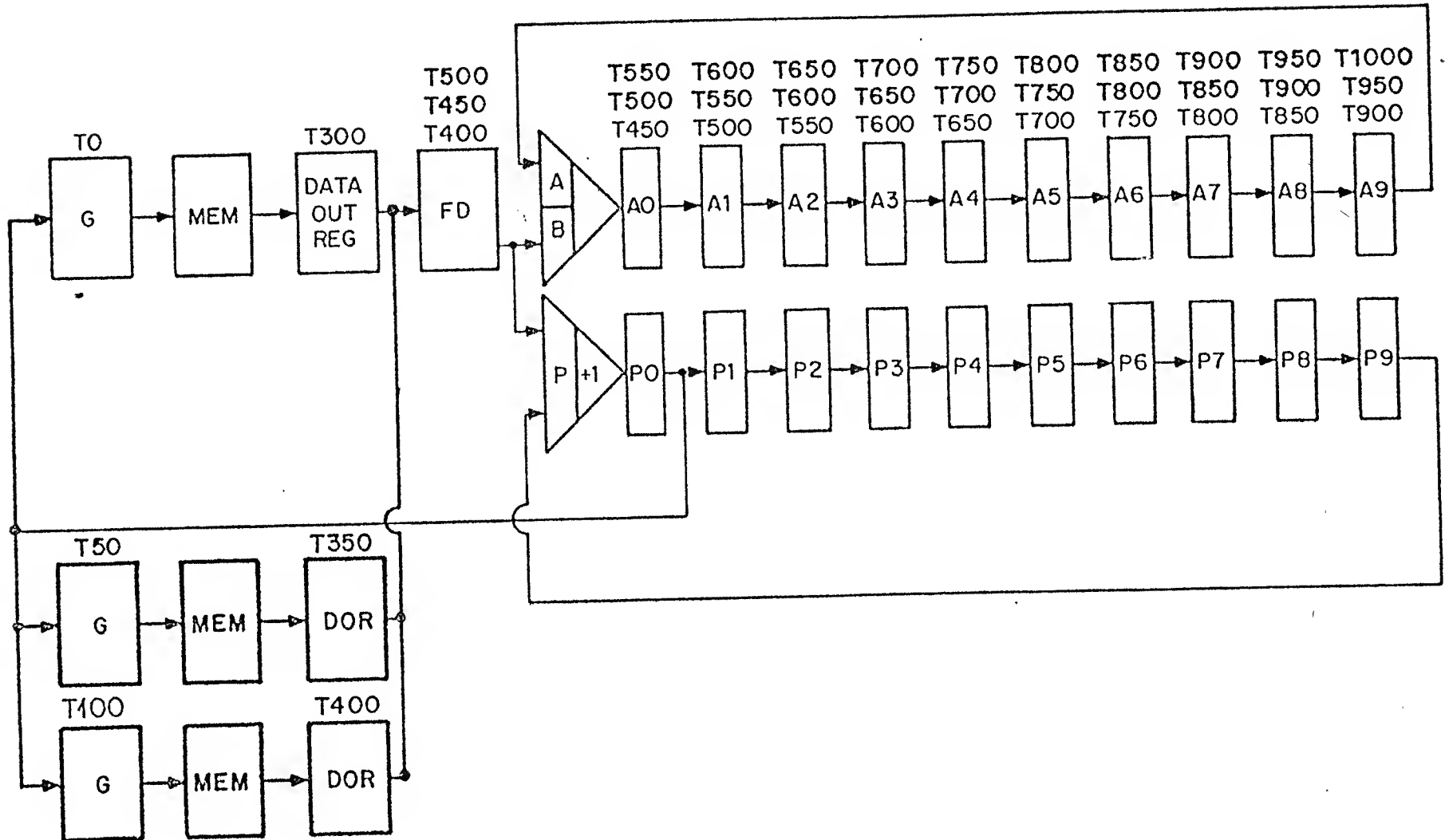
P7012

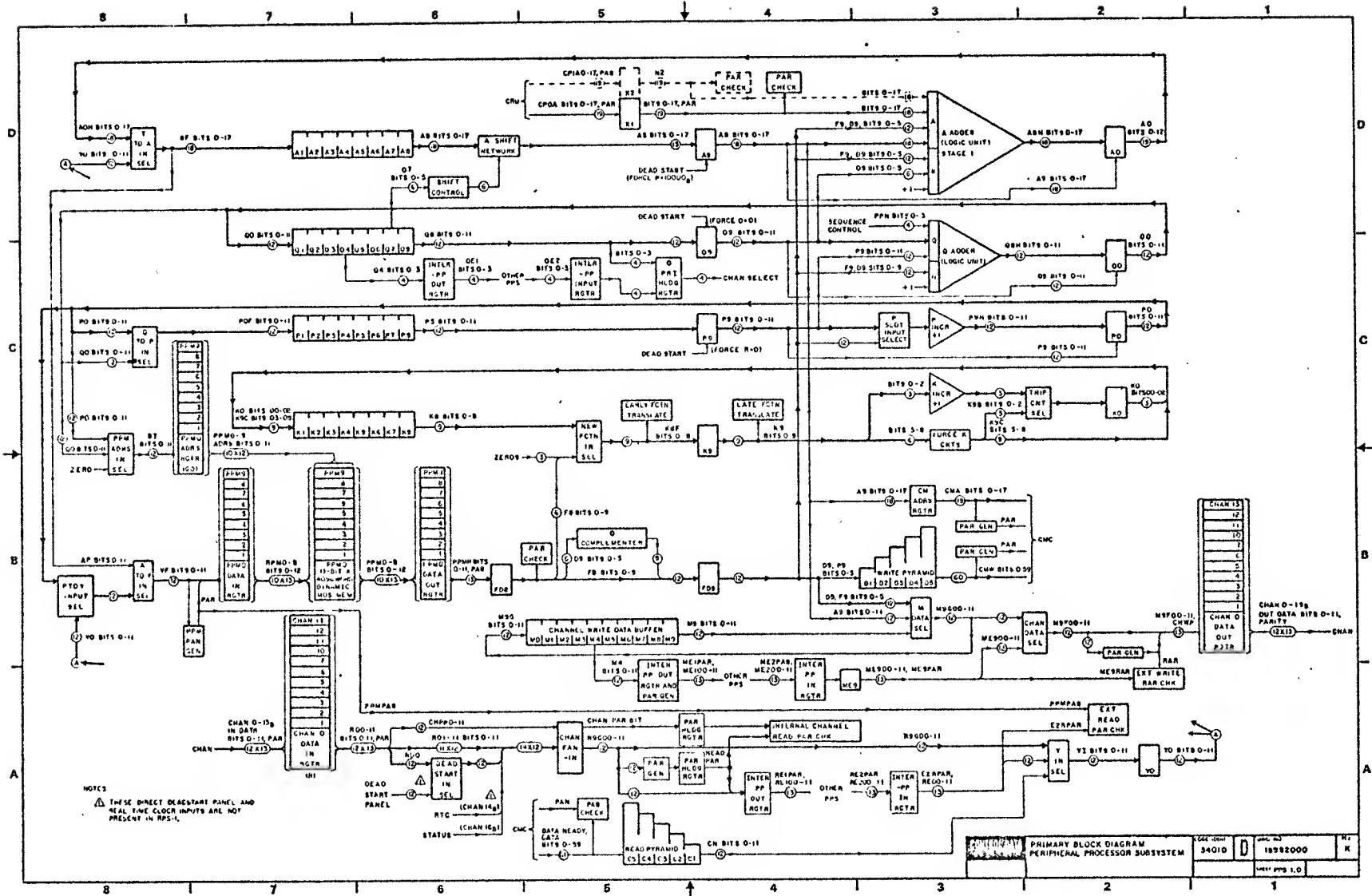
HO-13

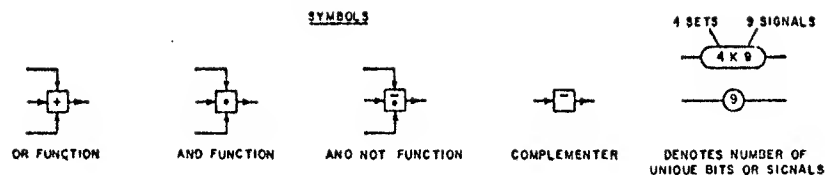
30

50

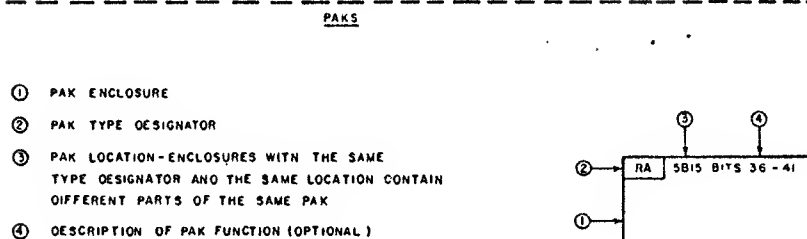
# BASIC PPU ( EXTEND A WITH SECOND AND THIRD PPU ) DIAGRAM





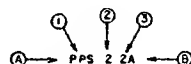


THESE SYMBOLS DENOTE THE FUNCTION BEING PERFORMED REGARDLESS OF THE TYPE OF GATE BEING USED IN THE LOGIC CIRCUIT. THUS, THE AND FUNCTION IS ENABLED BY COINCIDENT INPUTS AND THE OR FUNCTION IS SATISFIED BY THE PRESENCE OF EITHER INPUT

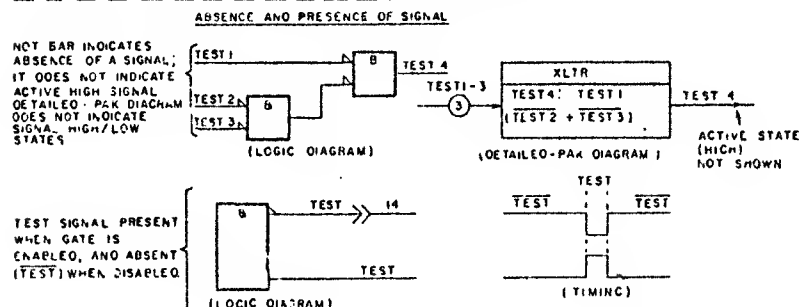
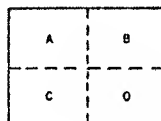


REFERENCES

A REFERENCE IS LOCATED AT THE BEGINNING OR END OF ANY LEAD THAT HAS ITS ORIGIN OR DESTINATION ON A PAK NOT READILY ACCESSIBLE TO THE LEAD. REFERENCES ARE MADE TO OTHER SHEETS WITHIN THE UNIT OR TO SHEETS IN OTHER UNITS. A REFERENCE CONSISTS OF THE FOLLOWING INFORMATION:



- ④ SHEET NUMBER - A SHEET NUMBER CONSISTS OF THE FOLLOWING PARTS:
- ① UNIT ABBREVIATION
  - ② FIGURE NUMBER - FIGURES ARE NUMBERED SEQUENTIALLY 1 X FOR THE PRIMARY BLOCK DIAGRAM, 2 X FOR THE SECONDARY BLOCK DIAGRAM, AND 3 X FOR THE DETAILED-PAK DIAGRAM.
  - ③ DRAWING NUMBER - DRAWINGS ARE NUMBERED SEQUENTIALLY WITHIN THE FIGURE, BEGINNING WITH ZERO
  - ④ LOCATION - QUADRANT OF REFERENCED SHEET TO FURTHER AID IN LOCATING THE REFERENCED POINT QUADRANTS ARE IDENTIFIED AS SHOWN



#### STACKED PAKS I/O CONVENTIONS

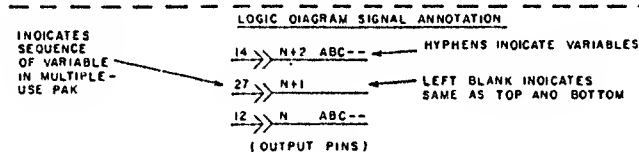
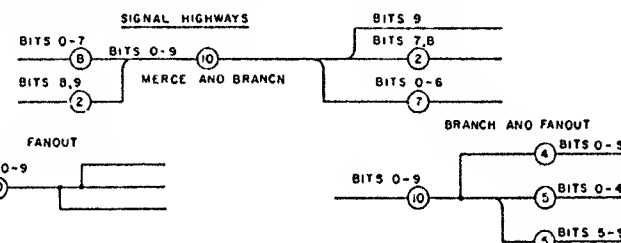
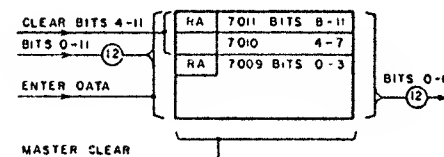
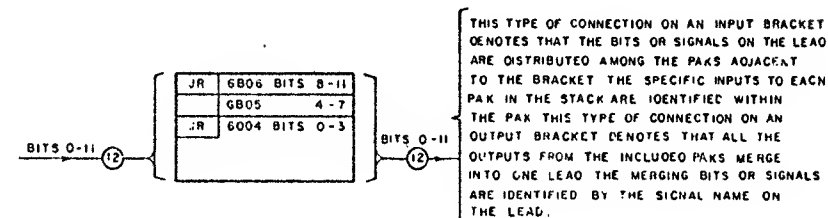
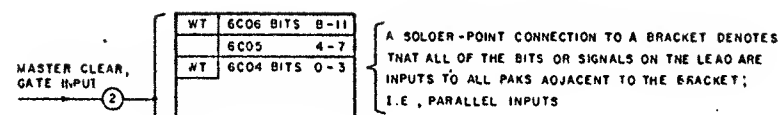
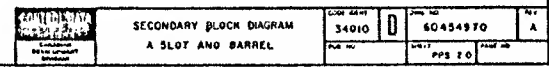


Figure 5-1-7. Key to Diagram Symbols



## PRIMARY BLOCK DIAGRAM (PPS 1.0)

A peripheral processor subsystem (PPS) consists of 10 peripheral processors (PPs), each having 4096 12-bit words (plus parity bit) of metal oxide semiconductor (MOS) memory. The PPs are coordinated by a multiplexing system that gives each of them access to common hardware for arithmetic, logical and I/O operations, as well as for communication with central memory (CM). One PPS controls 12 bidirectional I/O channels with all PPs having mutual access to all channels. Two PPSs (models 173/174 only) make available 24 such channels, with any PP being capable of addressing any channel in either PPS.

All PPs communicate with each other or with external equipment over the 12-bit (plus parity) channels. One channel handles only one external equipment at a time, but all 12 channels may be active at once.

Each PP does 64 instructions including:

1. Arithmetic operation
2. Logic
3. Input/output
4. Central memory read/write
5. Exchange jump

The I/O instructions of a PPS determine the status of a channel, select the channel, select the device, determine the device operation, and transfer the data to or from the channel.

The PPS exchanges data with CM via central memory control (CMC) by means of the CM read/write instructions. The read pyramid of a PPS receives 60-bit words from central memory and disassembles them into 12-bit PPS words which go to successive locations in peripheral processor memory (PPM). Conversely, a write pyramid assembles five PPS words to form a word for transmission to CM.

The PPS generates a parity bit on data or address transmission to CM and checks parity on data from CM. Parity generators also produce parity bits on all inputs to a PPM (peripheral processor memory) and on I/O data or function transfers. To indicate a parity error, the appropriate bits of the status and control register (SCR) are set with a different bit being reserved for each type of error.

Programs for all 10 PPs are stored in separate PPMs, but instruction execution hardware is time shared. To facilitate this sharing, data pertinent to PPs not using the hardware at a particular time is stored in the barrel which consists of a number of shift registers. A clock advances the data in these registers every 50 ns. The execution hardware is at the end of these registers. When the data of a PP is presented to the hardware, it is in the slot. The barrel is the set of shift registers which rotates the data of the different PPs in and out of the slot. The major registers in the barrel are the A, Q, P, K, and M registers.

When data is shifted out of the slot associated with one of the barrel registers, the register receives this data into its first flip-flop (rank 0). The register then shifts the data into its next rank on each successive clock pulse until reaching rank 9, which is the slot. Thus a trip around the barrel is completed every 500 ns, this being a major cycle. A minor cycle is the 50-ns clock period.

To allow software compatibility with the CYBER 70 series, a 1000-ns major cycle is required. A unit called the skip control circuit forces a 1000-ns major cycle time for a PP by inhibiting changes in the slot on every second trip around the barrel. An SCR bit sets the major cycle time of all PPs to either 500 ns or 1000 ns.

### MEMORY

The 10 PP memories are phased in conjunction with the data in the barrel so that each memory will be doing the same operation when its data is in a particular rank of the barrel. PPM reconfiguration switches on the dead start panel allow any PPM to be PPM0. PPM0 has a special controlling function at dead start time.

### A REGISTER

The A register (18 bits) holds one operand for arithmetic/logic operations, a CM address for CM communications, and word counts or data for I/O instructions.

In the slot, the A adder performs a number of arithmetic/logic operations. One of the several choices of operands in the A slot is the X register which contains the current central processor unit (CPU) address. There are two X registers for a two-CPU system.

On dead start, A is set to 10,000<sub>0</sub>.

### P REGISTER

The P register is the program address register. In the slot it increments, decrements or passes.

On dead start, P is set to zero.

### Q REGISTER

The Q register (12 bits) holds data for several functions including operand addresses, relative jump designator, shift count, the upper 6 bits of a CM address, and the channel number on all I/O and channel branch instructions.

In the slot, the Q adder adds, subtracts or passes.

Dead start forces each rank of Q to its corresponding PP number.

### K REGISTER

The K register holds the f (6-bit instruction code) portion of an instruction word and the trip count (3 bits). These 9 bits together determine the operations to occur on a particular instruction at a particular stage of completion.

The K incrementer and force K circuits in the K slot increment or force K to a new value upon completion of certain stages of an instruction.

On dead start: (load), K → 710  
(sweep), K → 505  
(dump), K → 730.

### INPUT/OUTPUT

Each channel has a single register that holds the data word being transferred in or out (although two separate registers are shown in PPS 1.0 for illustrative purposes). The maximum transfer rate is one word per 500 ns.

The state of a switch on each channel enables or disables the parity checking capability of a channel. In this way, parity checking is prevented for external devices with no parity scheme.

The A register accepts single word transfers from the channels, whereas block transfers of words are routed through the PPM data register to memory.

Both the PPS and external devices use full and active channel conditions to control the bidirectional information interchange through the channels. These conditions are indicated by active/inactive and full/empty flags. Active indicates that the channel is in use; full means that data is available in the channel data register.

One PP communicates with another by inputting data from a channel that was receiving output from another.

On dead start, channels 0 - 11<sub>g</sub> are assigned to corresponding PPs by loading the Q barrel positions with the appropriate channel numbers.

### PPS-1 DIFFERENCES

There are few differences between PPS-0 and PPS-1 at the PBD level. Since the PPS-1 real time clock cannot be read on a channel and PPS-1 has no dead start panel, there is no input to PPS-1 channel fan-in from either of these. RTC and dead start panel inputs are received by PPS-1 (as channels 14<sub>g</sub> and zero, respectively) via the inter-PPS output register.

## DETAILED PAK DIAGRAM (PPS 3.1)

### A BARREL

The A Barrel consists of an input selector, a shift register, an A shifter and controls, and a (A = 0, A = 1) test circuit.

#### INPUT SELECTOR

The BR module produces a signal BRYTOA which determines whether the input to the A barrel shift register will be Y (the data in register) or A0. The only time  $Y \rightarrow A$  is on a 70 instruction.

#### A BARREL SHIFT REGISTER

The A barrel shift register shifts A data from ranks 1 to 8 with no change in the contents for a particular PP. Shifts occur on the trailing edge of T1.

#### A SHIFTER AND CONTROLS

The A shifter is used on the 10X (SHIFTD) instruction. Its 5 ranks shift A by a number of positions that are successively higher powers of 2 (i.e., rank 1 shifts A one place; rank 2, two places;... rank 5, sixteen places).

The output of a particular rank X depends on its control bits, QSC5 and QSC(X - 1). QSC5 specifies an end-off right shift; if QSC5, a circular left shift occurs. QSC(X - 1) enables the shift action of rank X.

A particular rank causes left, right or no shift by gating an input bit to the appropriate output bit which would feed the next rank.

A SLOT

The zero filler bits shown entering all ranks are used for the end-off right shift where zeros must be forced into the resultant vacant positions.

The BLSHFT signal is produced on trip 1 of the 10 instruction (100 - NF) as a K8 translation. Two minor cycles later, when another instruction is in progress, BLSHFT enables to enable control bits QSC0-5.

The A shift count complementer (BE module) inverts Q7 bits 0-4 if enabled by Q7 bit 5.

Q705 is passed to the A shifter unchanged to indicate direction of shift, and Q7 bits 0-4 indicate the shift magnitude.

#### A = 0, 1 TEST CIRCUIT

The AE module produces the signals AGPIZ and AGP2Z indicating that AS bits 5-11 and 13-17 are all zero. The AU module uses these signals, combined with the remaining bits, to generate AEQ0 (A = zero) and AEQ1. These, in turn, indicate the end of a block transfer on I/O instructions.

#### DEAD START

The signal DSRT forces A912 to a one while all other A9 bits go to zero (shown on AX and AE paks - PPS 3.2). This sets A to 10,000<sub>8</sub> during dead start.

## DETAILED PAK DIAGRAM (PPS 3.0)

### A SLOT

The A slot is comprised of input selectors, an arithmetic/logic unit (ALU), a function code generator, and an output selector.

#### INPUT SELECTORS

In the BL module, selector signals are generated according to REFRF7 (early refresh signal), certain K8 conditions, and D800 (D8, bit 0). These signals determine what the two ALU inputs (A and B) will be.

BLA10,1 select the A operand from X1, X2, FD and A9. The 27, 36 and 37 instructions are the only ones that do not produce an  $A9 \rightarrow A$  code. On the 27 instruction, either  $X_1 \rightarrow A$  or  $X_2 \rightarrow A$  with D800 specifying which central processor address is read. On 36 and 37, the A slot control signals increment FD by forcing  $FD \rightarrow A$  and adding a 1 from the B input.

In the BL module, the signals FTOB, DTOB and QTOB are generated to select the B operand.  $B \leftarrow 1$  when none of the above signals is present.

The REFRF7 signal is present on rank 8 of the barrel during the time that PP0 is about to enter the slot on a memory refresh cycle. REFRF7 inhibits any changes that may occur during early translation. In the A slot, REFRF7 prevents any changes on the contents of A.

#### FUNCTION CODE GENERATOR

The BM module produces function codes according to K8 and NF8. In the absence of any of the codes specified in detailed pak diagram PPS 3.0, BM produces an  $A \leftarrow AB(00000)$  code. In this way, or by disabling the ALU output, A can remain unchanged in the slot.

On the generation of the AMINUS B control signal, the function code generator also produces the SUBTR signal. This ensures that the two unused input bits in the AH pak (location F38) are forced to ones to facilitate a ones complement subtraction within the pak.

#### ARITHMETIC/LOGIC UNIT

The ALU can perform 31 operations. However, the function codes provided in the A slot make use of only eight of these.

The A adder carry-summing circuit produces 4 group carry bits (AGCY 1-4) in parallel by using generate and propagate (AGPG, AGPP) bits from each of the five groups. Carry bit 3 (A1CY03) is used as the 5th group carry bit. This partial look-ahead addition method provides greater speed than ripple-through carry propagation methods.

#### OUTPUT SELECTION

The signal BJA B enables the A adder output (AB), allowing it rather than A9G to pass on to A911.

## DETAILED PAK DIAGRAM (PPS) 3.2)

### Q SLOT

The Q slot is comprised of input selectors, an arithmetic logic unit (ALU), a function code selector, and an output selector, and an output selector.

#### INPUT SELECTION

The BN module generates input control codes for both operands used in the ALU. One of the operands, H, is selected from D, FD or 1 (one). H = 1 when neither DTOH nor FTOH are enabled.

The Q slot input selector (AG) determines the Q operand according to the conditions DSRT (dead start) and PTOG.

The significance of the Q barrel in channel selection and the nature of Q input at dead start time are as follows:

In a 10-PP system, Q bits 0-3 are used to select one of channels 00-16<sub>10</sub> on I/O instructions. For a dual PPS system, there are two such sets of channels, one in each chassis.

A fifth bit, Q bit 4 specifies which chassis is involved. Q bit 4 = 1 specifies a channel in the other chassis (an external channel). Q bit 4 = 0 specifies an internal channel.

At dead start time, a sequence of 10 PP numbers (0 - 9) is fed into rank 9 of the Q barrel from the PP code counter. Each rank of the Q barrel is thereby loaded with a number one higher than the previous rank.

Because Q designates the channel to be used on I/O instructions, this process of loading Q900-04 effectively assigns each PP to a channel. PP n is defined as the PP assigned to channel n at dead start time.

In a dual PPS system, a scheme allowing each PPS to be assigned to the other's channels at dead start is provided. Reversing the channel select switch on the dead start panel activates the DSPPSW signal at PPS-0, thereby setting Q904. The signal is relayed to PPS-1 in the form of DSPSEL and E2DSQ4, where it clears Q904. Since Q904 = 0 for PPS-1 and 1 for PPS-0, each PPS is assigned to the other's channels. This feature is only enabled in a full 20-PP system.

Hardware involved in carrying out these functions is outlined in detailed pak diagrams 3.2 and 3.3. For a description of how the PPS interprets Q register bit 4 after it is loaded in the slot, see figure 4-2-1 and other explanatory material relating to PPS 3.3.

#### Q ADDER FUNCTION CODE GENERATOR

The Q adder function code generator (BN) produces either one of the three function codes shown in detailed pak diagram PPS 3.2 or the Q (00000) code. REFR7 produces a Q code and inhibits PTOG, thus causing a pass in the Q slot prior to a refresh cycle.

#### LOGIC UNIT/ADDER

The Q ALU carries out five functions: passing H, passing Q, adding (Q plus H), subtracting (Q minus H), and incrementing (Q plus H where H = 1).

The Q adder carry-summing circuit (AU) uses group generate and propagate bits (Q GP 0-2 G, P) to produce carry bits in parallel, and thus speeds up the addition process.

#### OUTPUT SELECTION

BPQHTQ (from BP pak) determines whether or not the Q adder results (QH) will be used.

## DETAILED PAK DIAGRAM (PPS 3.3)

### Q BARREL

The main components of the Q barrel are the Q barrel shift register, QTOG shift register, and channel select circuitry.

#### Q BARREL SHIFT REGISTER

As shown in detailed pak diagram PPS 3.3, the Q barrel shift register is divided into two parallel paths: one for the lower 6 bits and the other for the upper 6 bits.

The lower 6 are used in I/O instructions with 0-3 forming the channel number. Bit 4 is the chassis designator, and bit 5 is an escape bit. This escape bit causes an I/O instruction to exit rather than hanging up the PP involved. For instance, if Q205 is present an exit will be generated on an activate (74) instruction when the channel is already active. At rank 8, the Q = 0 test circuit (AU) produces the signal Q8EQ0 which is used to detect the end of a block transfer to or from central memory.

#### CHANNEL SELECT CIRCUITRY

The EZ module transmits the contents of the inter-PP channel select output register (the channel address) to the other PPS Q pri selector (AF pak). If the receiving PPS has accepted an external request, the signal BWEREQ occurs, thereby causing the external channel number (QE2, bits 0-3) to go to the channel select code holding register. Here the BD modules generate the CHER 00-13 and SEL 00-13, 16 signals which are channel designators for parity errors and I/O respectively.

Bit 4 of the Q barrel has special significance in channel operations. There are two sets of channels in a dual PPS system, one in each PPS chassis. On I/O instructions, bit 4 is the part of the channel address designating which set is being referenced. Regardless of which PPS is doing the I/O instruction, when bit 4 is loaded into the slot Q904 = 1 means PPS-1 is being referenced and Q904 = 0 means PPS-0 is being referenced.

It follows that Q bit 4 = 1 in PPS-0 must force a reference to an external channel, whereas the same condition in PPS-1 refers to an internal channel. Therefore, the two PPSs interpret Q bit 4 differently. This difference is illustrated in figure 4-2-1 where Q804PR in PPS-0 is simply a fan-out of Q804, while the PPS-1 Q804PR is an inverted Q804. In both PPS chassis, Q804PR is used exclusively as an external channel indicator. Thus, Q804 = 1 will designate a PPS-1 channel in either chassis because PPS-0 and PPS-1 would interpret it as an external and internal channel indicator, respectively.

Figure 4-2-1 also shows how PPs of either PPS may be assigned to the other's channels at dead start time. When the channel select switch is on during dead start, the DSPPSW signal is activated. This forces Q9G04 to 1 in PPS-0 and has the opposite effect in PPS-1. As a result, the PPs of PPS-0 are assigned to channels of PPS-1, and vice versa.

Figure 4-2-1 shows a functional representation of the paks involved in the manipulation of Q bit 4. Internally, the paks used in both PPSs are identical. DPDs 3.2 and 3.3 show how wiring changes force identical paks to different functions.

In PPS-0, the circuitry shown in PPS 3.3 enables the selection of external channels in a 20-PP system only. For a 10-PP system, Q24DIS = 0 and this feature would be disabled.

#### QTOG SHIFT REGISTER

The QTOG shift register holds a QTOG flag (BYQTOG) which is generated on conditions as shown in the G input control translator (detailed pak diagram 3.8). If the next major cycle is a skip, BYQTOG (called Q to G9S at Q to G shift register, rank 9) regenerates the QTOG signal at the G input control translator. This is done because on the skip cycle the conditions causing QTOG (i.e., NF9) may be absent. In the absence of the BYQTOG flag, the P → G condition would arise.

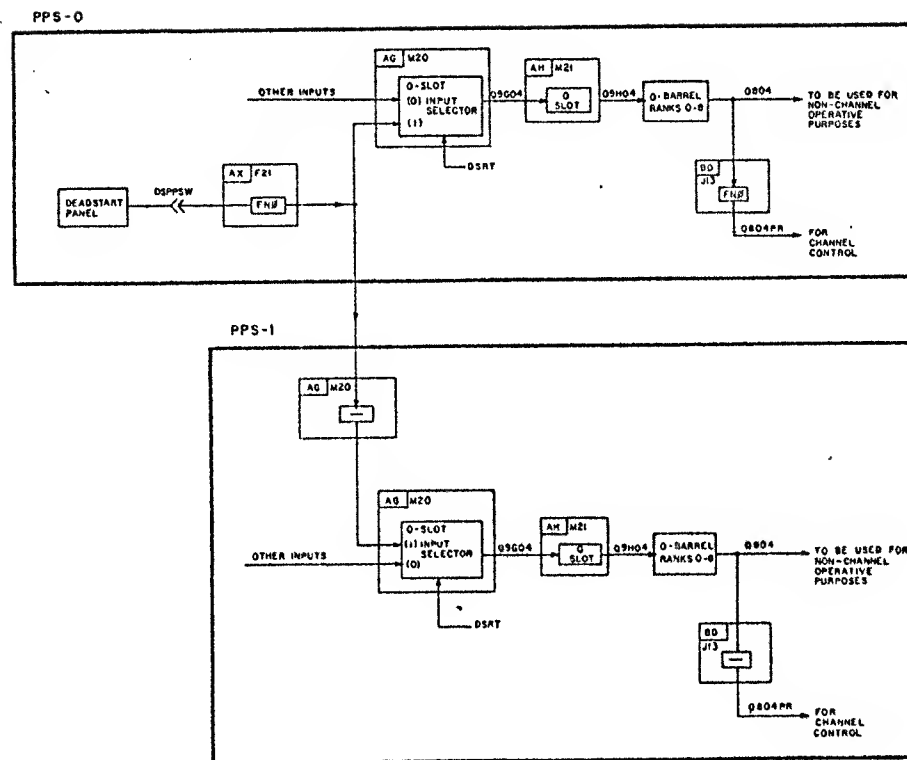


Figure 4-2-1. Bit 4 of Q Register

DETAILED PAK DIAGRAM (PPS 3.4)  
P SLOT AND BARREL

The P slot is comprised of an input selector, a function control, an incrementer and an output selector.

The P barrel is a shift register with an input selector.

P SLOT INPUT SELECTOR

In the AF module, the state of BUFDTP gates either P9 or F9D9 to the incrementer. The P slot control translator translates K8 conditions and the state of FLAG1 to form the FDTP signal.

FUNCTION CONTROL

When the A slot control translator produces the signal BLPINC, the P incrementer increments; otherwise, it decrements.

FLAG18 generates both the BLFDTP and BLDEC signals on K=612 to cause FD-1 → P. The BQ module then resets FLAG1 to ensure that FD-1 → P occurs only on the first 612 trip of a 61 instruction.

P INCREMENTER

Depending on the condition of PINC, the P incrementer either increments or decrements.

Carry out bits, PCY 3, 7 from the two lower order groups of the incrementer are used directly as carry in bits. No look-ahead carry-summing circuit is used.

The carry in bit for the lowest order group (bits 0-3) is a zero. No end around carry-bit (PCY 11) is necessary because P is always positive.

SLOT OUTPUT (OR BARREL-RANK 0 INPUT) SELECTOR

The P0 input selector gates either P9G or the output of the P incrementer (PI, bits 0-3) to rank 0 of the P-barrel. BJPI, which is produced in the A/P control translator (BJ), enables the P1 input.

BARREL AND BARREL INPUT SELECTOR

QTOP controls the barrel input selection. If QTOP, then P0 goes to the P barrel shift register, ranks 1-8. Q → P on all jump instructions for which the jump conditions are satisfied. In these cases, Q is the jump address.

## DETAILED PAK DIAGRAM (PPS 3.5) K SLOT AND BARREL, AND FUNCTION TRANSLATIONS

Detailed pak diagram PPS 3.5 shows the components of the K slot and barrel.

Ranks 0-7 of the barrel are comprised of a shift register in which the contents shift unchanged from end to end. Rank 8 of the barrel does K8 function decoding and new function input selection.

The main components of the K slot (rank 9) are function translators, K incrementer, force K circuitry and a K0 input selector. The lower 3 K bits are the trip count, and the upper 6 form the function (or instruction) code.

### K BARREL

Ranks 0-7 of the K barrel (AJ module - PPS 3.5) form a simple shift register.

In rank 8, the new function input selector either passes the function code and trip count already in the barrel, or gates in a new function code with a zero trip count. NF8 is the control level that causes the latter selection to be made. It comes from rank 8 of the new function register (AE module - PPS 3.5). The input to this register is EXIT, a signal generated at rank 9 of the previous cycle on the last trip of an instruction. Thus the K barrel uses an instruction termination signal (EXIT) to initiate a new instruction.

Before a PP enters the slot, the K8 decoders (PPS 3.5) perform early translations in preparation for slot time. For instance, the A, P and Q registers use these translations to gate inputs to their slots.

### K SLOT - FORCE K CIRCUITRY

The KINCR signal gates bits 0-3 from the K incrementer into K0. If KINCR is a zero, the trip count from the force trip count decoder enters the barrel.

The force trip count decoder and translator generates a trip count according to the progress of the ongoing instruction, and presents it to the K0 input selector.

This forced change in trip count, along with a forced change in function code produced in the force K function decoder from bits BCB 0-2, allows a complete change in a PP instruction.

One instance of a forced change occurs on K = 5x1. In this case, the K9 decoder generates the level BY3YY (Y indicates that these bits will stay the same - i.e., 3x1 will be forced). This is translated in the force K-FCTN translator to produce  $\overline{BCB2}$ , BCB1,  $\overline{BCB0}$  (code 2) which results in 3x emerging from the force K function decoder. Likewise, the force trip count decoder would force the trip count to stay at one.

### K 9 DECODERS AND TRANSLATORS

The K9 decoders and translators use the K9 bits as well as the NF9 and skip conditions to produce control signals used throughout the PPS.

In the exit control translator (BX - PPS), the EXIT signal is generated when conditions indicate the end of an instruction. The BX module uses channel condition levels (BXINT, FULL, ACTV, EXFULL, EXACTV) as well as K9 translations to produce EXIT 2 as an indicator for I/O instructions. All other exit conditions are produced in the BV module and are indicated by the signal BVEXIT.

A special exit condition is the CXEXIT signal. By means of settings in the STC register or by manual switches, CXEXIT is generated on the slot time of a particular PP. The result is that any PP can be forced to exit regardless of other conditions.

The BV module also produces BVKI, a signal which is used along with other conditions in the BK module to produce the KINCR signal.

Note that all signals prefixed by EX (other than EXIT) are status indicators pertaining to the channels of the other PPS.

DETAILED PAK DIAGRAM (PPS 3. 6)  
CHANNEL IN AND PPM WRITE DATA

Detailed pak diagram PPS 3. 6 shows a series of input selectors that determine the PPM data input.

AG modules comprising stages 1 and 2 of channel fan-in determine which of channels 0-16<sub>g</sub> will be presented to the Y input selector. The control bits for this purpose, QPR 0-3, are the actual channel numbers from the Q barrel.

Note that the channel data in/out register handles data transfers in both directions. The same register is shown as it is used for data output in detailed pak diagram PPS 3. 10.

The Y input selector chooses from external channel data (RE00-11), internal channel data (R9G00-11), or read pyramid data (CX00-11). The word selected may go to the A barrel or be written in PPM. Control bits for this selection, BRY10, 2 originate from the Y Input select bit translator in the BR module.

Also in the BR module is the Y control translator that generates BRPTOY and BRATOY. These signals determine the outputs of the P → Y and A → Y input selectors, which are the last two stages of the Y input (PPM data input) selection.

The Y control translator also generates the BRPPMW (peripheral processor memory write) and BRCHPC (channel parity check) control bits.

BRPPMW goes to PP memory control to force a memory write cycle. If  $\overline{\text{BRPPMW}}$ , then PPM control forces a read cycle. In other words,  $\overline{\text{BRPPMW}} = \text{READ}$ .

PARITY

On a read from internal channels, the parity error signal BFCHST originates from the BF pak (location G29). The enabling levels ENBLPC, FULLTO and ENCHPC ensure that a parity error could only be signalled on a data channel read instruction for which the peripheral device should generate parity.

Parity checking for external channel reads is carried out in the BF modules in locations (G29, N14).

The parity bit (PPMPAR), which is stored in PPM with each data word regardless of its source, may be forced to zero by STC081.

## DETAILED PAK DIAGRAM (PPS 3, 7)

### PPM DATA PATHS

Data enters and leaves the PPM through the same CC modules.

On a write instruction, data enters the data input register from the Y register upon gating by SEL (select memory). It is then presented to memory by DIG (data in gate) which only occurs on a write instruction.

The read/write amp (CU or CS) gates data from the data in register to the I/O terminals of the MOS memory on a write instruction. The control signals for the operation are CHENM (chip enable) and WRENM (write enable). A high CHENM signal activates the read/write amp and a low WRENM enables the data transfer path from data in register to PPM.

Each of the 10 PP memories is organized into four groups of 1024 words each. The group being addressed on a memory reference is selected by control signals CLKMn0-n3, where n is the PP number. Only one of the four CLKM signals presented to each PPM is active at one time in order to select one of the four memory groups. The PP memory control generates a 10-bit address (ADMn 2-11) which selects a specific word within the selected group.

The reset (RST) signal serves to prepare address and data lines for a PP memory reference.

The sequence of events for a normal (no refresh) memory cycle is shown in detailed pak diagram PPS 3, 7.

The sequence of events for a read cycle is very similar to that of a write cycle. SEL occurs twice on a read cycle, but only once on a write. The first occurrence is used only on a write. The second gates data from memory to the data out register after which DOG (data out gate) presents the data to FD7. Since WRENM and RDENM are opposite signals, both generated from HREAD, only one exists during a particular memory cycle. A low RDENM signal that exists on a read cycle, gates data from MOS memory through the read/write amp to the data out register.

The CHENM0 signal on a read cycle differs from that on a write cycle only in that it starts 50 ns later.

The generation of control signals, phases (PHA 0-9) and addresses are shown in detailed pak diagrams PPS 3, 8 and 3, 9.

Note that the MOS memory logic levels are produced by means of ECL/TTL conversion, followed by TTL/MOS conversion.

# DETAILED PAK DIAGRAM (PPS 3.8) MEMORY ADDRESS PATHS

The address input selector (PPS 3.8) determines which of Q0, P0 or zero will be the PPM address according to control signals QTOG and ZTOG. G is the address holding register. In the absence of QTOG and ZTOG, P-G. Control signals are produced in the BS, BY, BT and AE modules in PPS 3.8.

A special case is the QTOG signal. Its generation may be dependent on the state of NF9. NF9 only exists on the first major cycle following a skip. If a PP is operating at 1X speed, each trip requires two major cycles, and the NF9 signal will be lost on the second cycle. The Q-G flag, QTOG9S is produced from the signal BYQTOG which occurs on every function decode requiring Q-G. If the next cycle is a skip, the presence of QTOG9S enables BYQTG causing Q-G once again. If QTOG was not enabled on this second trip, P-G. Timing for Q to G9S is shown in figure 4-2-2.

The 10 address holding registers (one for each PPM) of the CJ module gate 10 of the 12 address bits through ECL/TTL converters to the MOS memories. Multiplexing signals PHA 0-9 select the correct PPM address register according to which PP is accessing memory.

If a refresh cycle is present, AIG = HREF will occur, causing CLRA (refresh address) bits 7-11 to become bits 3-7 of the PPM address.

The address holding registers of the CE module decode the other 2 bits of the PPM address to select a quadrant of the PPM being addressed. Here again, there is one such register for each PPM, one of which is selected on every minor cycle according to the PHA 0-9 signals. The quadrant to be reset (precharged - see PPS 3.7) is selected in the same way. On refresh, the HREF signal occurs, causing all four quadrants to be selected for a particular PPM.

This addressing method used on refresh, addresses 1/32 of PPM on every refresh cycle by addressing one column of the PPM address matrix each time.

Refresh timing circuitry is described in detailed pak diagram PPS 3.9. Timing for the PHA, AIG and HRST signals is shown in PPS 3.7.

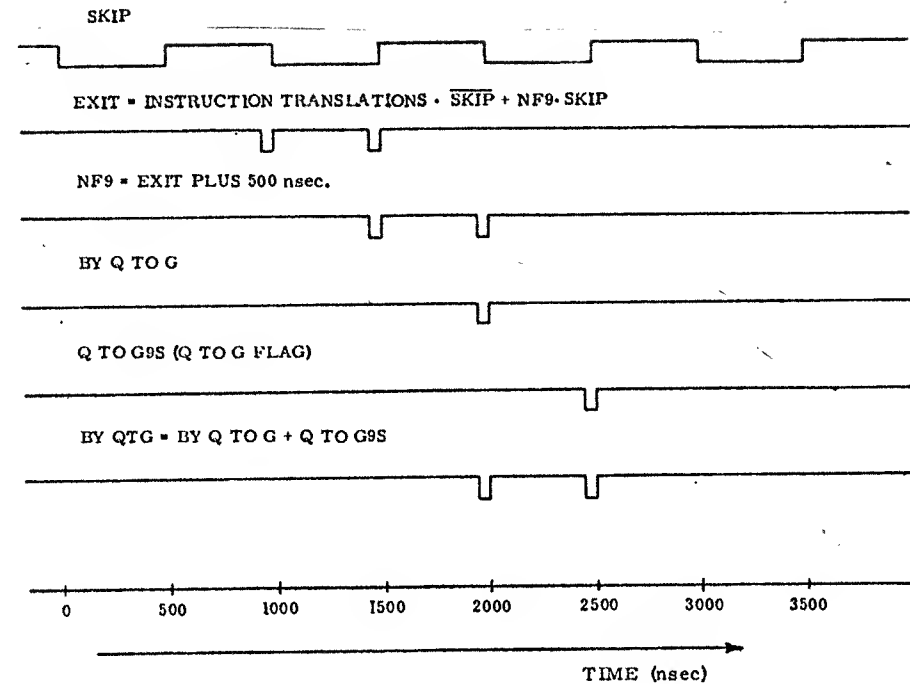


Figure 4-2-2. Q to G Flag

## DETAILED PAK DIAGRAM (PPS 3.9)

### PP MEMORY CONTROL

#### MEMORY PHASES

The memory sequence decade counter and memory sequence decoder produce memory timing signals PHA 0-9 (phases 0 to 9).

As an example of the synchronizing effect of the PHA signals, observe the generation of the HREAD 0-9 signals (PPS 3.9). The Y control holding register (PPS 3.6) passes the read signal to the read flip-flop of PP memory control on rank 0 of the PPU. Assuming PHA0 exists at this time, read forces an HREAD0 output (the signal that causes a read in PPM0) at the read flip-flop. On the next minor cycle, PHA1 exists and the read signal produces HREAD1, setting up PPM1 for a read. In successive cycles HREAD2, HREAD3, HREAD4, ... etc., are produced. This process continues until the PP that generated HREAD0 once again appears. Again, HREAD0 will occur on the appearance of this PP. All memory control signals are produced in this way, with each memory being controlled sequentially by different phases, so that the PPs are synchronized with the PPMs.

The time relationship of the memory phases and the ranks of a particular PP for PPM0 are shown on detailed pak diagram PPS 3.7.

A fixed timing relationship exists between the PP count and the major cycle signal, AXMJC. This relationship is utilized in the CG pak to allow the assignment of any PPM to a given PP by means of PPM select switches on the dead start panel. The setting of these switches is

the value to which AXMJC resets the memory sequence decade counter. As a result, the memory phase signals PHA0-9 may be advanced relative to the PP count by any desired number of minor cycles.

This PPM reconfiguration capability is disabled in PPS-1 on 14 and 17 PP systems. The memory sequence decade counter always presets to zero in these systems. Only in a full 20PP system are the PP select switches enabled in PPS-1. Reconfiguration in PPS-1 occurs in parallel with that in PPS-0. As a result, the two PPSs have identical reconfiguration assignments.

#### REFRESH CONTROLS

Shown on PPS3.9 are the refresh timing and address counter circuits. The refresh timing circuit enables the CL 32 signal once every 32 $\mu$ -sec during the odd major cycle. Any PP executing a PPM read cycle will be forced to refresh during the CL32 'on' time. PPMREF causes a skip cycle to occur after the refresh period because no data is read during refresh. A timing diagram showing the major signals related to the refresh cycle is on PPS 3.9.

When CL 32 occurs on a read cycle, the memory reference is postponed until the skip cycle which follows refresh time. However, a write memory cycle would go on uninterrupted while setting the delayed refresh FF. This flip-flop then forces a refresh on the skip cycle following the write.

## DETAILED PAK DIAGRAM (PPS 3.10)

### PPM READ DATA OUTPUT

#### PPM READ PARITY CHECKER

T1 strobes data bits PPMR0-11 from PPM into rank 8 of the FD register.

The PPM parity checker (PPS 3.6) compares the parity bit from memory (PPMRPB) with one it generates itself from D8, F8 bits 0-5 and produces a parity error bit (BFPMSPE) if they differ. BFPMSPE forces a 507 instruction if allowed by the SCR, with the result that the offending PP hangs. Fifty ns after BFPMSPE, a 17-ns pulse, BFPMSPE goes to a fan-in circuit that selects one of PMER00-12<sub>g</sub>. PMER00-12<sub>g</sub> in turn sets a bit in the SCR indicating which PPM was in error.

#### D8 COMPLEMENTER

The D8 complemeter inverts D8 on the relative jump instructions (03-07) if D8 bit 5 = 1. On these instructions, D is added to the present address in Q to arrive at the new address. Since all negative numbers are in ones complement form, D8 bit 5 indicates a negative number. In order to carry out the addition of this negative number to the present address, the Q adder subtracts the inverted D8 bits from Q.

#### D= 0 TEST CIRCUIT

The D = 0 test circuit (PPS 3.10) produces the signal DEQZR0 which is used on the K increment and G input control translators to skip trip 2 of instructions 01, 02, 5X

#### M REGISTER

FD9 is used in many locations throughout the PPS since all data from memory goes through the FD register. The M input selector may gate FD9 directly to the channel data I/O register depending on the state of BJMI bits 0, 1 and BWPR.

When a PPS is addressing its own channels, FD → M on every cycle of the 732 instruction. P, which holds the address of the next word to be read, is not incremented unless the empty response is received signalling that data has been accepted by the channel.

On an external write, no response comes from the other PPS until after the data is sent. If Flag II = 1, P will increment and A, the word count, will decrement. When the BF1FUL response occurs, a new word enters M.

Flag II prevents a new word from entering M by recirculating the first word in the M register until a BF1FUL response is received. Entry of a new word into M at this stage would be undesirable because P has already incremented with the result that the wrong word is in FD. Further changes in the A and P registers are also inhibited by Flag II.

Similarly, transmission of an incorrect function code caused by a P increment on the first 770. NFF trip is prevented by Flag II which causes recirculation of the correct code in the M barrel until the code is accepted by the other PPS.

ME9 → M whenever channel control accepts a request from the other PPS, as indicated by BWPR1. If a read request from the other PPS generates BWPR1, then channel control (detailed pak diagram PPS 3.12) would not generate the control signals such as AQXM to allow transmission from the channel in/out register (PPS 3.10). Channel control prevents transmission of any unwanted outputs from M by not generating the transmission signals AQXM 00-11. This includes the cases when ME9 → M on a BWPR1 arising from an external read request or on a 70 instruction with a BF1FUL response.

Note that A → M on a 72 instruction when a CZEFUL or EXHPLY response is received. This indicates that the other PPS did not accept the write request and, therefore, the same data from A must be output once again.

TABLE 5-A-1. PPS KEY EQUATIONS

SIGNAL	T. P.	CODE	SOURCE	PAGE	TRANSLATION
<b>'A' ADDER INPUT</b>					
A → A		3	BL	3.0	00-28+30-35+40-77*REFR7
ED → A		2	BL	3.0	REFR7(36-37)
N <sub>1</sub> → A	G27.2	1	BL	3.0	27 REFR7, D800
N <sub>2</sub> → A	G27.2	0	BL	3.0	REFR7, D800, 27, CPIENA
D → B	J38.2		BL	1.0	(1xx+30x+35x+2xx)REFR7
F → B	J38.1		BL	4.0	(10x+35x+2xx)REFR7
Q → B	J38.1		BL	3.0	2xx+REFR7
-1 → B			BL	1.0	(0xx+35x+77x)REFR7+REFR7
A → 0	I25.14		AU	3.1	
A → 1	I4.7		AU	3.1	
<b>'A' ADDER FUNCTIONS</b>					
AQB	G25.11	11	BL	3.0	11x+(23x+33x), NF8
A → B		4	BM	3.0	12x+(22x), NF8
A → B		10	BM	3.0	13x
B	G25.12	5	BM	3.0	14x+(20x+30x), NF8
B	G25.1	12	BM	3.0	15x
A PLUS B	G25.6	31	BM	3.0	16x+NF8, (21x+31x+35x+36x)+614+637
A MINUS B		26	BM	1.0	17x+NF8, (32x+37x)+712+732
<b>'A' ADDER BYPASS</b>					
A → A <sub>0</sub>	J36.5		BJ	3.0	[BMF712(FULL, ACTV, BWINT+ERPLY, EXFU, Q904)+BMF732(FULL, BWINT, ACTV+ERPLY, Q904, EXAC, EXFU, ALQ0+Q904, EXAC, FL119)+ (BMF712, BMF732)], SKIP, (BGA0+637)
<b>'A' REG INPUT</b>					
Y → A	I20.12		BR	3.1	70x, NF9
SHIFT → A	G27.3		BL	3.5	10x
<b>'Q' ADDER INPUT</b>					
P → Q	I116.9		BN	3.2	(03x-07x)REFR7
D → H	I116.1		BN	3.2	01x-021+03x-600+605-614+615+617+62, NF8+63x+635-64-77+REFR7
F → H	I116.3		BN	3.2	NF8(01x-021+03x-600+605-614+616+617+63x+635+64-77), REFR7
-1 → H	I116.4		BN	3.2	022+(601-604)+615+(NF8-62x)+635
<b>'Q' ADDER FUNCTIONS</b>					
Q PLUS H		31	BN	3.2	(011+021+022+5x1+(03x-07x), 0805+601-604+NF8-62x), REFR7
Q MINUS H		26	BN	3.2	(03x-07x)D805+615+635], REFR7
H		5	BN	1.2	(NF8-63x-07x)+010+020+4xx+5x0+610+630+(64-67x), REFR7
<b>'Q' ADDER BYPASS</b>					
Q → Q <sub>0</sub>	N21.5		BP	3.2	(0xx-61x+62x, BINWEN+63x, BIN3FULL+64-77x), SKIP
<b>P, P, M ADDRESS INPUT</b>					
Q → C			BY, BS	3.8	BVEN [62x+(60x+600, NF9)+3xx+(NF9+600, BGA0)+DN+RO(010+020+5x0)+(021+4xx+5x1)+NF0(610+630)+BNQTOP, QTOG9S, SKIP
Z → G	J21.6		BY, DT	3.8	610, NF9+610, NF9+710, NF9+730, NF9+SKIP, NF9+735+741, SKIP
P → G					(Q → G), (Z → G)
<b>'Y' REG INPUT</b>					
P → Y	J20.			3.6	(DSRT, EVEN)
C <sub>x</sub> → Y	I20.7		BR	3.6	+SKIP, (021+610, NF9+630, NF9+71x, NF9+73x, NF9)
B <sub>x</sub> → Y			BR	3.6	SKIP(601-604+614-617+BRCSF(600, NF9+613))
H → Y			BR	3.6	Q904, SKIP, (70x, NF9+712, ERPLY, (EXFU+EXAC))
A → Y	I20.2		BR	3.6	(C <sub>x</sub> → Y), (B <sub>x</sub> → Y)
<b>P, P, MEMORY WRITE</b>					
P, P, WHIL	I20.11		BR	3.6	SKIP(410, NF9+34x, NF9+34x, NF9+541+35x-37x)
<b>CHAN. PAR. CHECK</b>					
CHTC			BR	3.6	+35x-37x+410, NF9+541+BRCSF(600, NF9+613)+601-601+610, NF9+614+630, NF9+71x, NF9+712+BWINT, (ACTV+FULL), ERPLY, Q904, (EXFU+EXAC))+73x, NF9
<b>'M' REG INPUT</b>					
A → M		3	BJ	3.10	SKIP [712, (BWINT, (ACTV+FULL)+Q904, ERPLY, (EXFU+EXAC))+70x, NF9]
M9 → M		1	BJ	3.10	K903, FL119(EXFU, ERPLY)
FD → M		0	BJ	3.10	K903(EXFU, ERPLY+FL119)

## PPS INSTRUCTION SUMMARY

This instruction summary lists mnemonic code, address field, instruction name, and instruction length for each PPS instruction. More detailed information may be found in the CDC CYBER 170 Hardware Reference Manual, publication number 60420000.

Symbols used in the summary are:

- d 6-bit operand or operand address
- f 6-bit instruction code
- m 12 bits used with d to form an operand or operand address

These three words or subwords are drawn from memory and used in various combinations as instruction codes for the two PPS instruction formats (12-bit and 24-bit).

The 12-bit format (figure 5-A-1) is a standard PPM word broken into an operand address (d) and a function code (f).

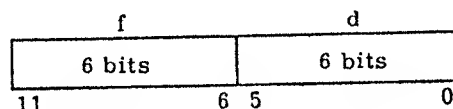


Figure 5-A-1. 12-bit Format

The 24-bit format (figure 5-A-2) uses the contents of two consecutive PPM locations (P and P+1). Here, P contains f and d while P+1 carries the m portion of the instruction. In this format, f is the instruction code and d, in combination with m, specifies further instruction parameters.

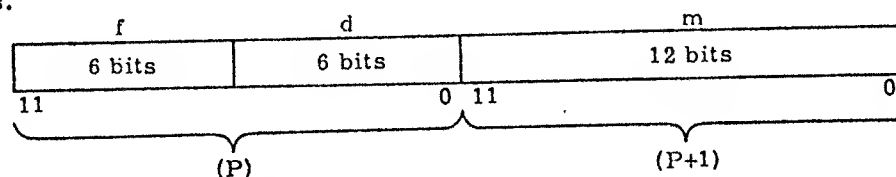
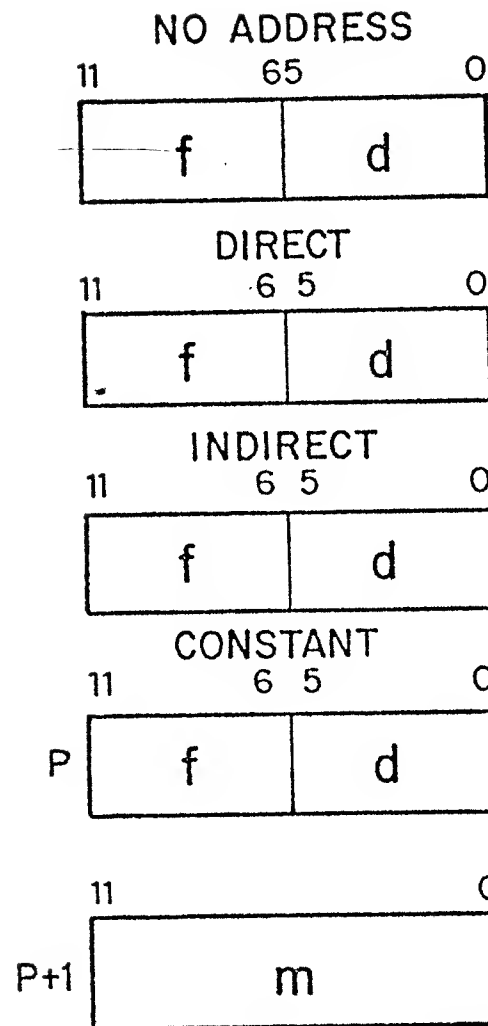


Figure 5-A-2. 24-bit Format

- A A register
- P Program address register
- Q Q register
- (x) Contents of a register or location x
- ((x)) x specifies an address whose contents is the address of an operand (indirect addressing).

# PPU INSTRUCTION & OPERAND FORMAT



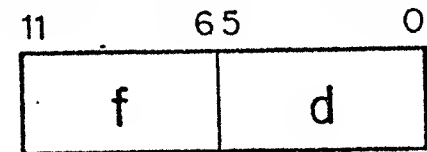
OPERAND=d

OPERAND=(d)

OPERAND=((d))

OPERAND=dm

## INDEXED DIRECT ADDRESS



OPERAND=  
o(M+(d))

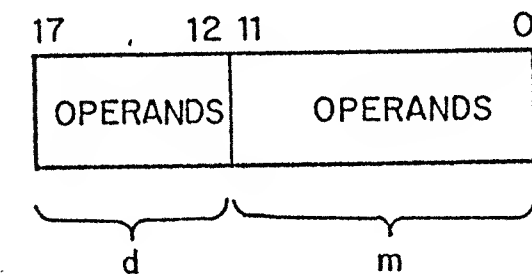
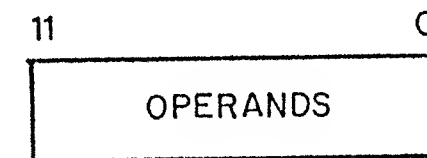
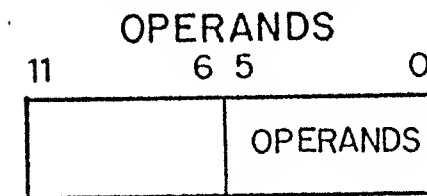
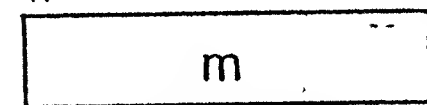


TABLE 5-A-2. PERIPHERAL AND CONTROL PROCESSOR INSTRUCTIONS

Octal Code	Mnemonic Code	Address Field	Instruction Name	Instruction Length (bits)
00	PSN		Pass	12
01	LJM	md	Long jump to m + (d)	24
02	RJM	md	Return jump to m + (d)	24
03	UJN	d	Unconditional jump d	12
04	ZJN	d	Zero jump d	12
05	NJN	d	Nonzero jump d	12
06	PJN	d	Plus jump d	12
07	MJN	d	Minus jump d	12
10	SHN	d	Shift d	12
11	LMN	d	Logical difference d	12
12	LPN	d	Logical product d	12
13	SCN	d	Selective clear d	12
14	LDN	d	Load d	12
15	LCN	d	Load complement d	12
16	ADN	d	Add d	12
17	SUN	d	Subtract d	12
20	LDC	dm	Load dm	24
21	ADC	dm	Add dm	24
22	LPC	dm	Logical product dm	24
23	LMC	dm	Logical difference dm	24
24	PSN		Pass	12
25	PSN		Pass	12
260	EXN		Exchange jump	12
261	MXN		Monitor exchange jump	12
262	MXN		Monitor exchange jump to MA	12
263	MXN		Monitor exchange jump to MA	12
27	RPN		Read program address	12
30	LDD	d	Load (d)	12
31	ADD	d	Add (d)	12
32	SBD	d	Subtract (d)	12
33	IMD	d	Logical difference (d)	12
34	STD	d	Store (d)	12
35	RAD	d	Replace add (d)	12
36	AOD	d	Replace add one (d)	12
37	SOD	d	Replace subtract one (d)	12

Octal Code	Mnemonic Code	Address Field	Instruction Name	Instruction Length (bits)
40	LDI	d	Load ((d))	12
41	ADI	d	Add ((d))	12
42	SBI	d	Subtract ((d))	12
43	LMI	d	Logic difference ((d))	12
44	STI	d	Store ((d))	12
45	RAI	d	Replace add ((d))	12
46	AOI	d	Replace add one ((d))	12
47	SOI	d	Replace subtract one ((d))	12
50	LDM	md	Load (m + (d))	24
51	ADM	md	Add (m + (d))	24
52	SBM	md	Subtract (m + (d))	24
53	LMM	md	Logical difference (m + (d))	24
54	STM	md	Store (m + (d))	24
55	RAM	md	Replace add (m + (d))	24
56	AOM	md	Replace add one (m + (d))	24
57	SOM	md	Replace subtract one (m + (d))	24
60	CRD	d	Central read from (A) to d	12
61	CRM	md	Central read (d) words from (A) to m	24
62	CWD	d	Central write to (A) from d	12
63	CWM	md	Central write (d) words to (A) from m	12
64	AJM	md	Jump to m if channel d active	24
65	IJM	md	Jump to m if channel d inactive	24
66	FJM	md	Jump to m if channel d full	24
67	EJM	md	Jump to m if channel d empty	24
70	IAN	d	Input to A from channel d	12
71	IAM	md	Input (A) words to m from channel d	24
72	OAN	d	Output from A on channel d	12
73	OAM	md	Output (A) words from m on channel d	24
74	ACN	d	Active channel d	12
75	DCN	d	Disconnect channel d	12
76	FAN	d	Function (A) on channel d	12
77	FNC	md	Function m on channel d	24

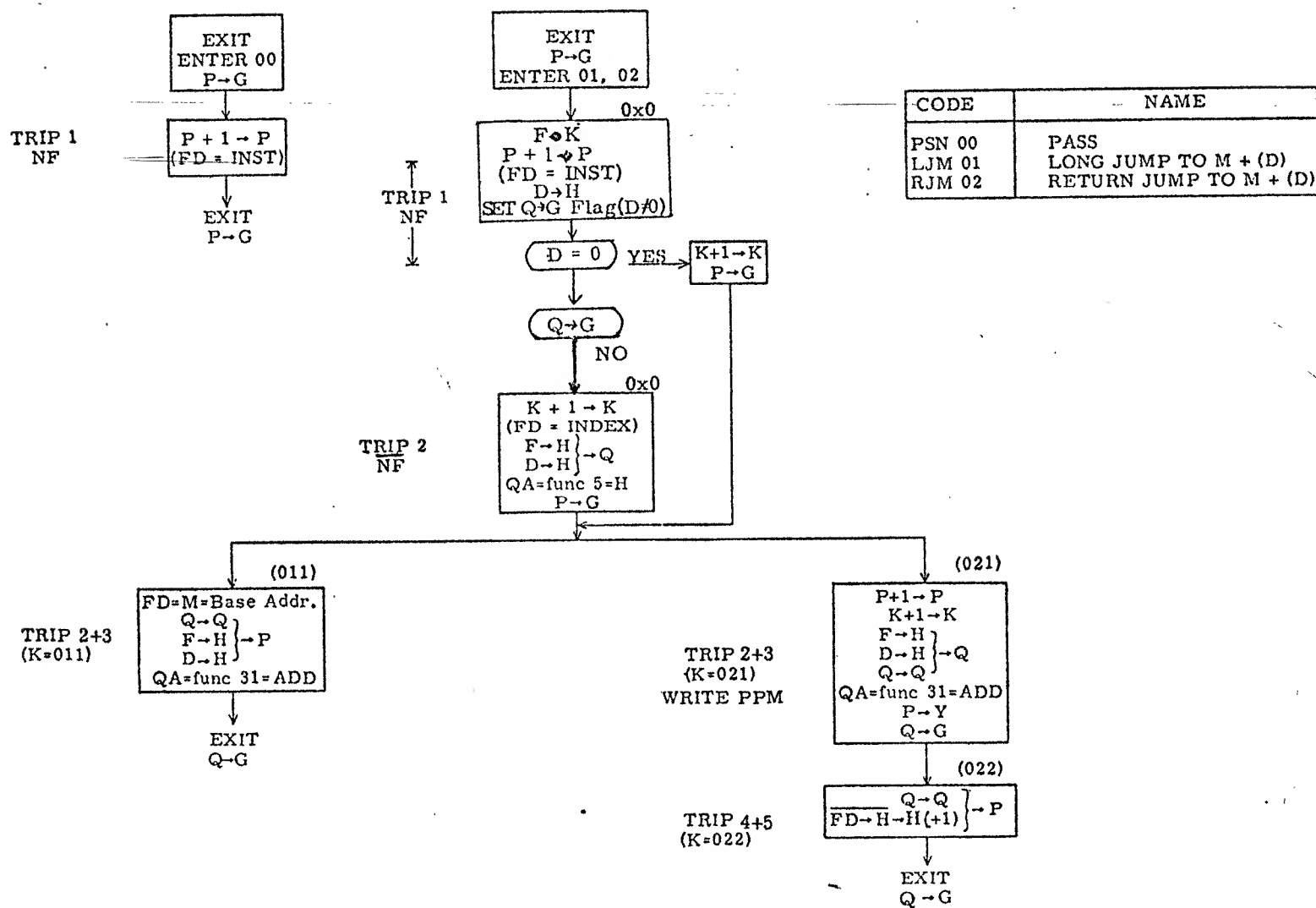


Figure 5-A-3. PPS Instructions 00, 01, and 02

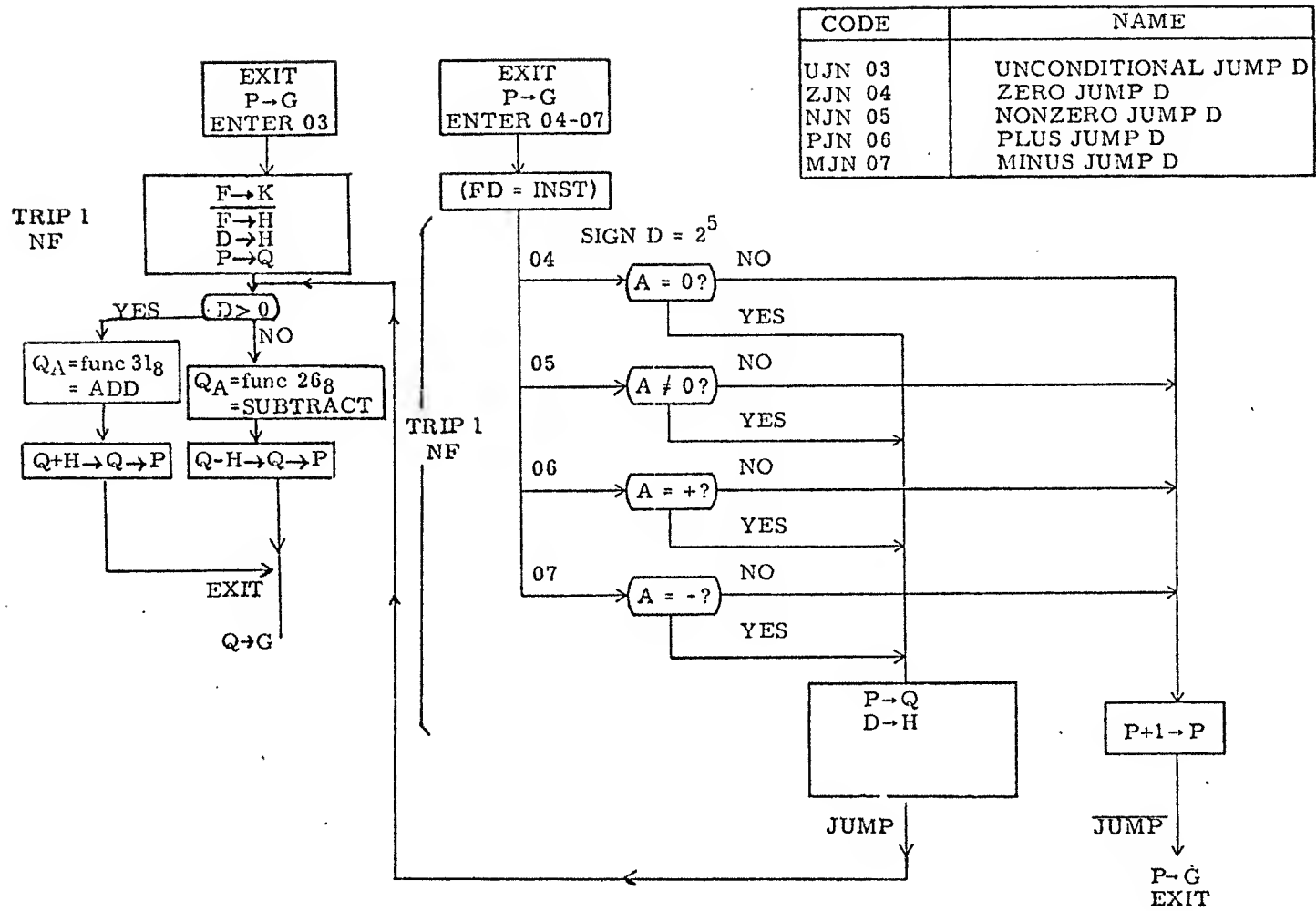


Figure 5-A-4. PPS Instructions 03 through 07

CODE	NAME
SHN 10	SHIFT D
LMN 11	LOGICAL DIFFERENCE D
LPN 12	LOGICAL PRODUCT D
SCN 13	SELECTIVE CLEAR D
LDN 14	LOAD D
LCN 15	LOAD COMPLEMENT D
ADN 16	ADD D
SEN 17	SUBTRACT D

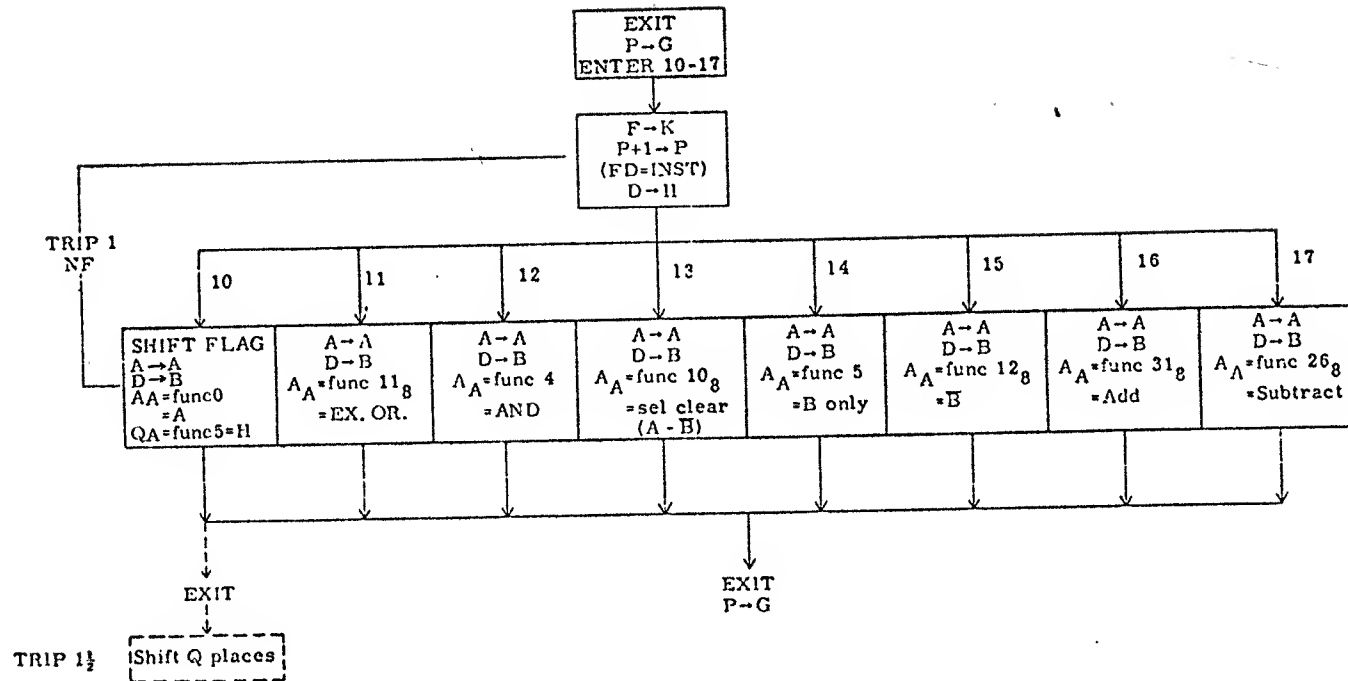


Figure 5-A-5. PPS Instructions 10 through 17

CODE	NAME
LDC 20	LOAD DM
ADC 21	ADD DM
LPC 22	LOGICAL PRODUCT DM
LMC 23	LOGICAL DIFFERENCE DM

① D = UPPER 6 BITS OF CM ADDRESS

② M = LOWER 12 BITS OF CM ADDRESS

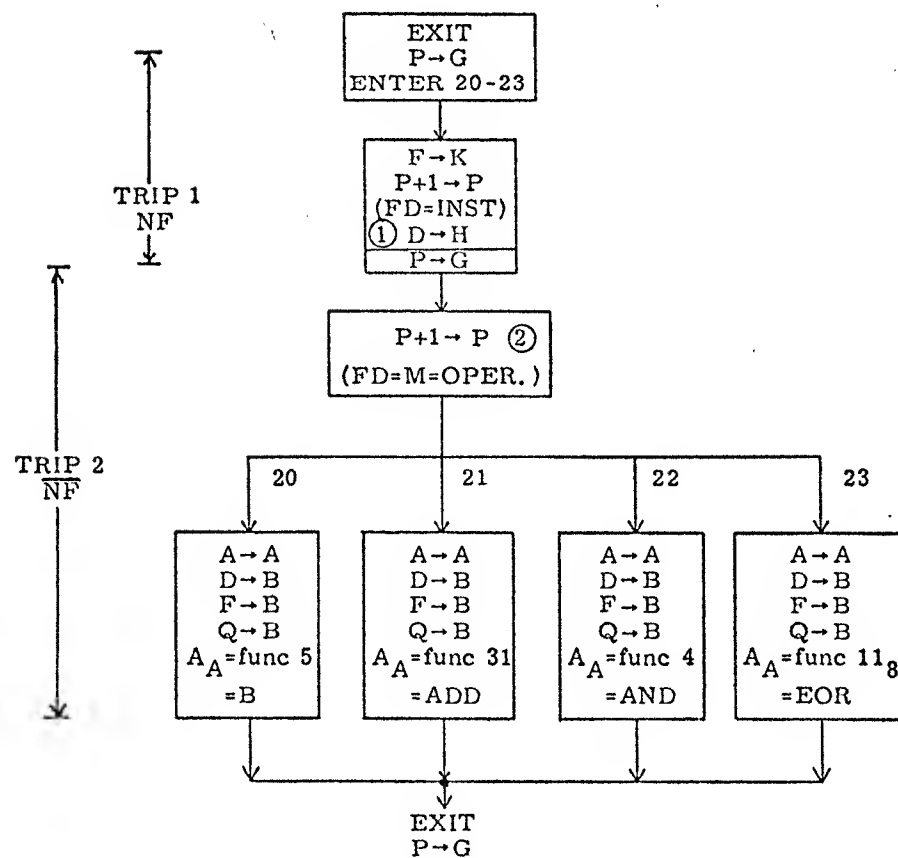


Figure 5-A-6. PPS Instructions 20 through 23

CODE	NAME
EXN 260	REGULAR EXCH.
MXN 261	MON. EXCH. A.
MAN 262	MON. EXCH. MA.

CODE	NAME
PSN 24	PASS
PSN 25	PASS
EXN 26	EXCHANGE JUMP
RPN 27	READ PROGRAM ADDRESS

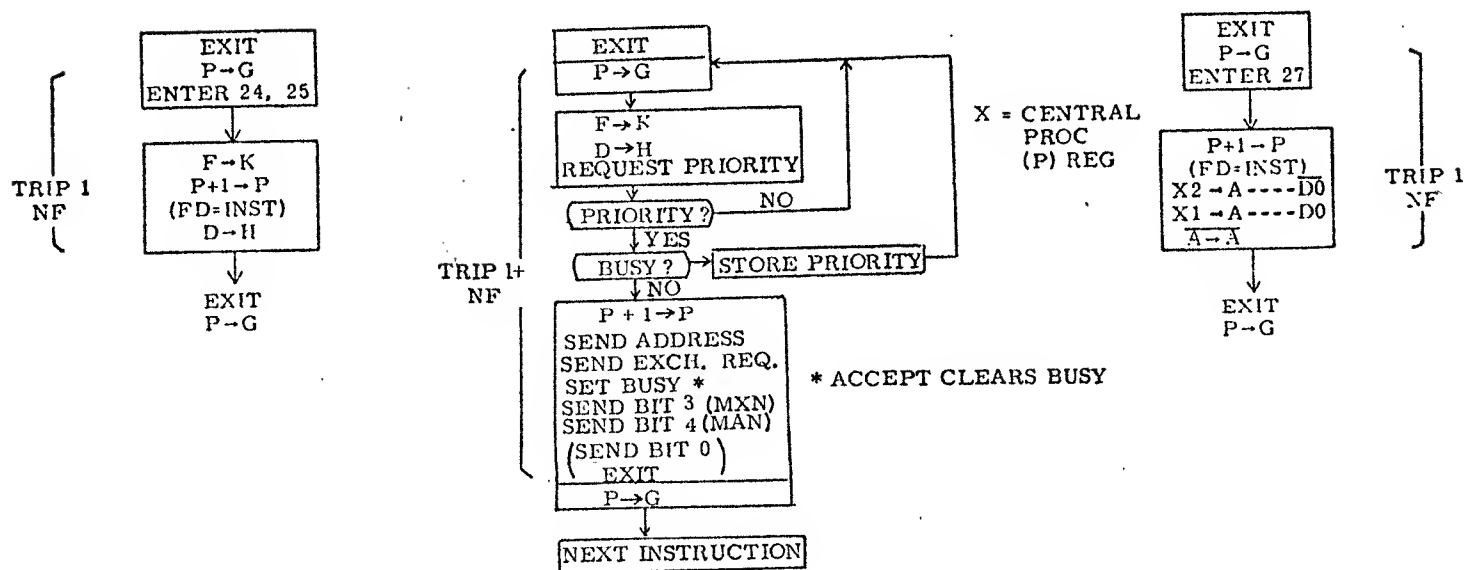


Figure 5-A-7. PPS Instructions 24 through 27

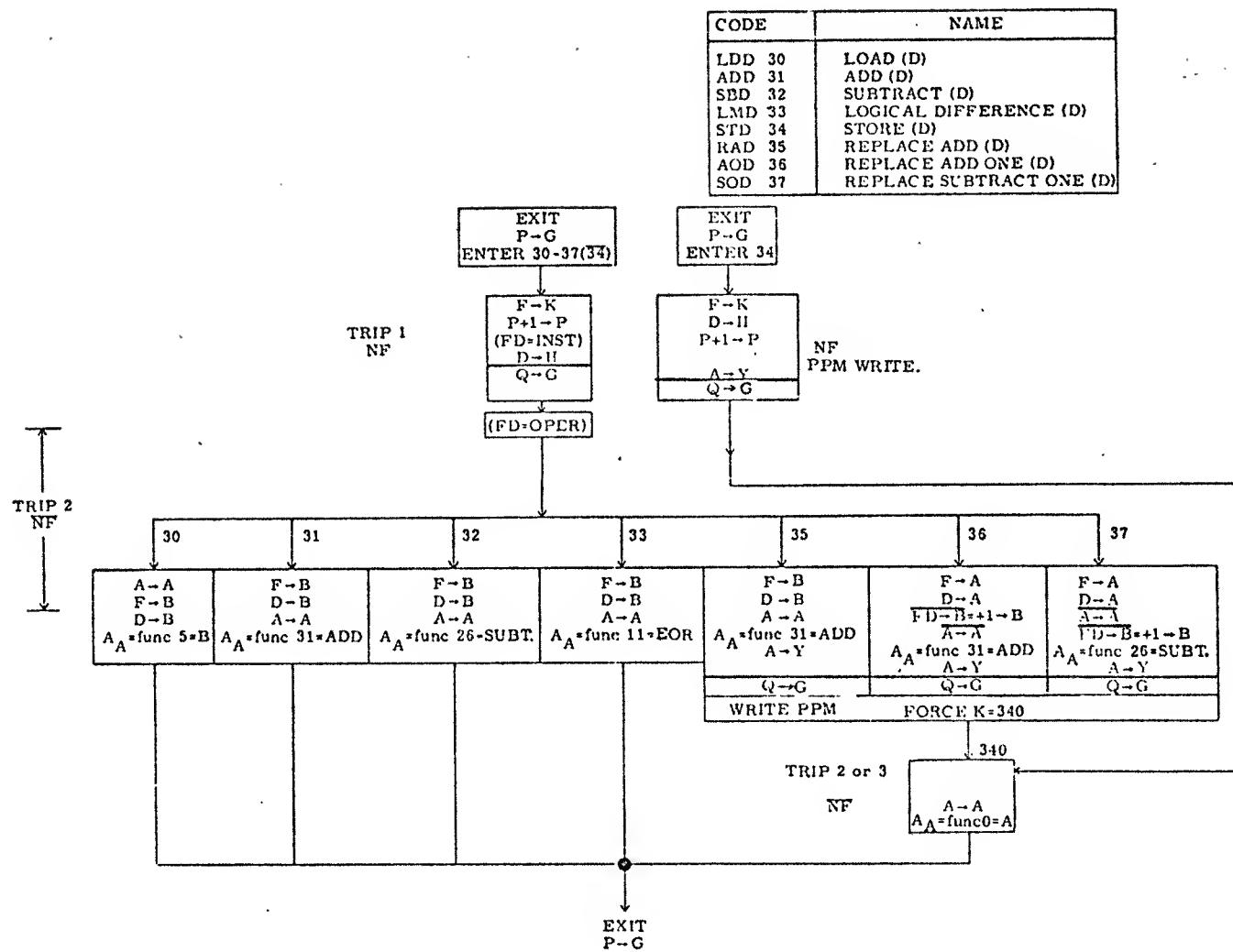


Figure 5-A-8. PPS Instructions 30 through 37

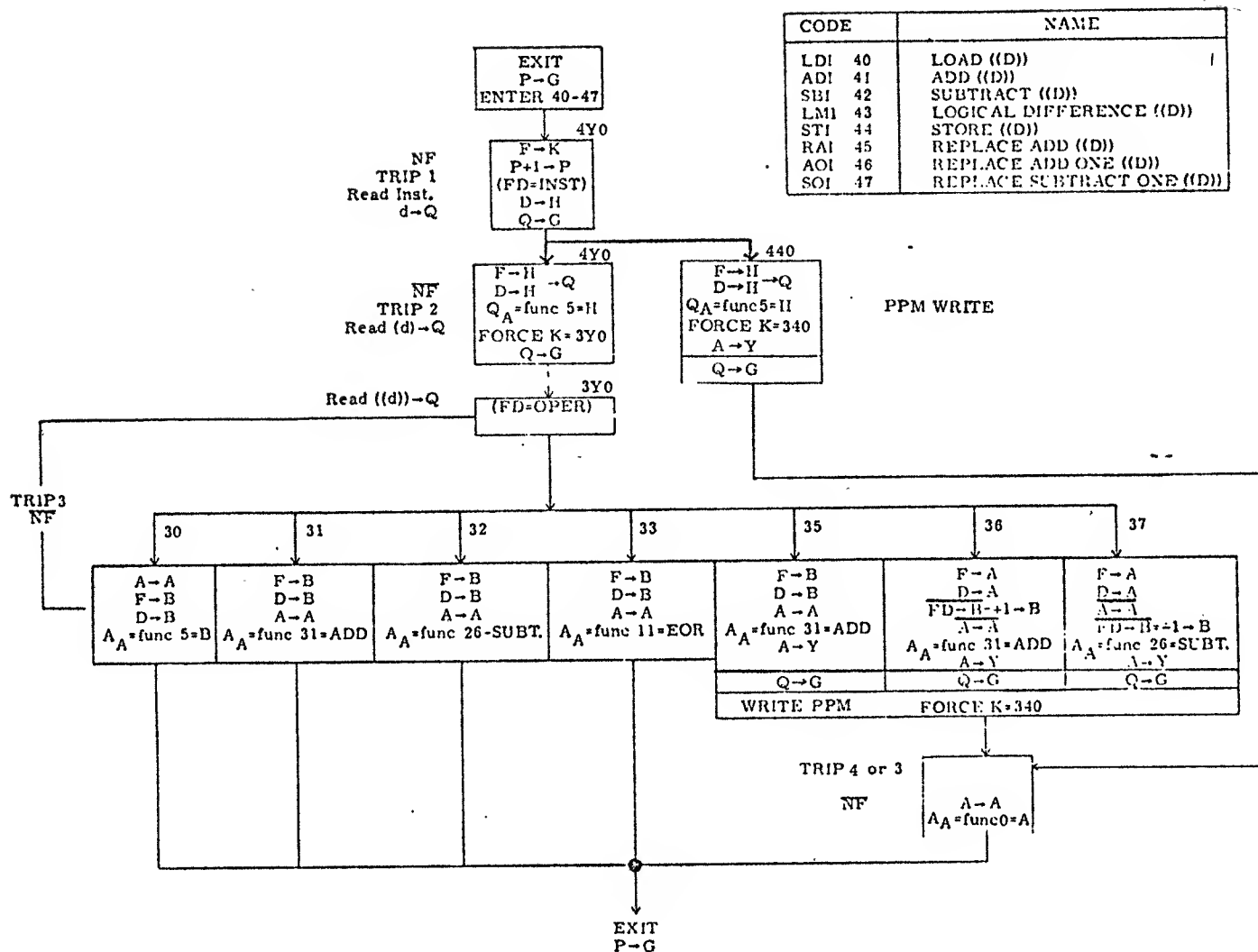


Figure 5-A-9. PPS Instructions 40 through 47

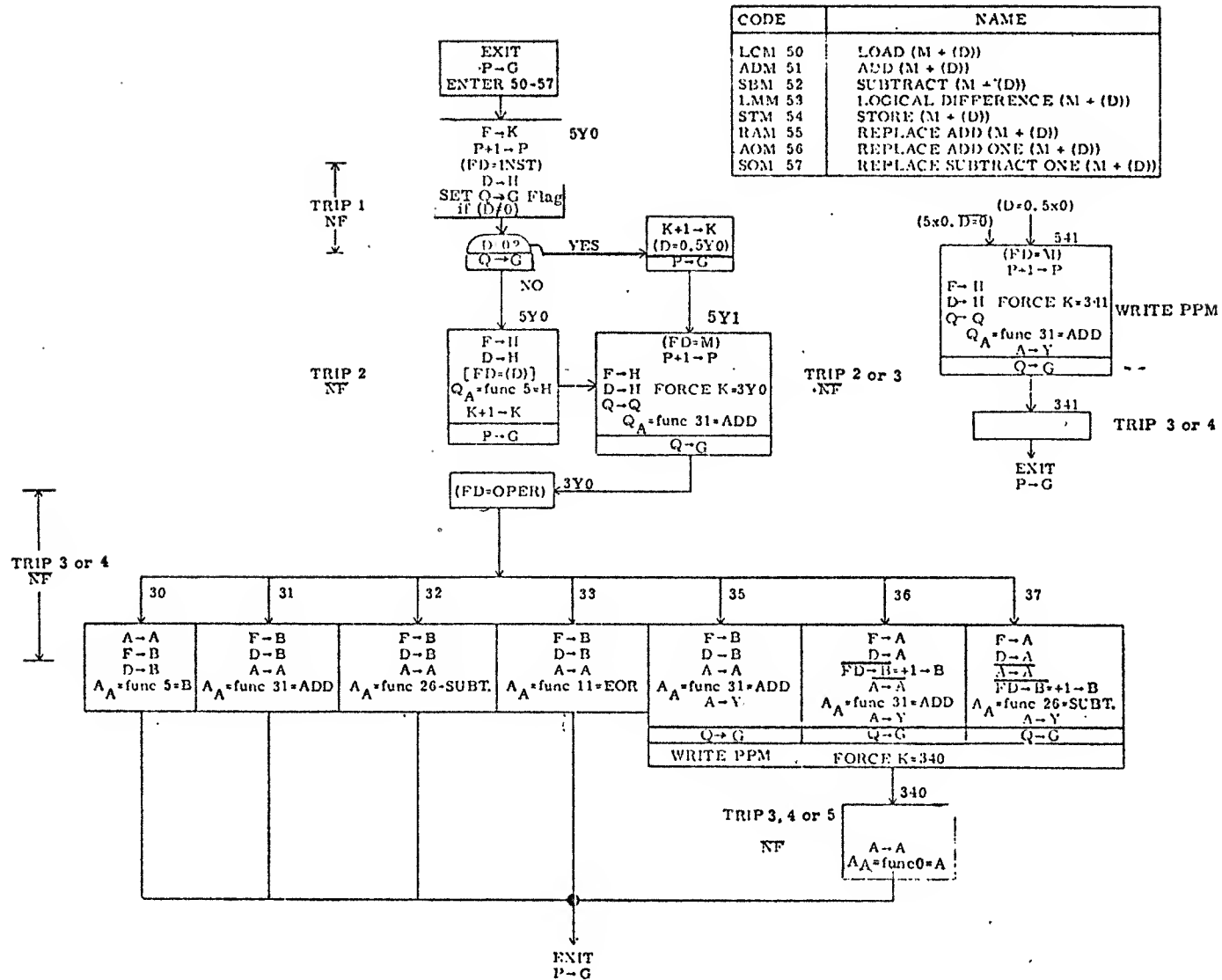
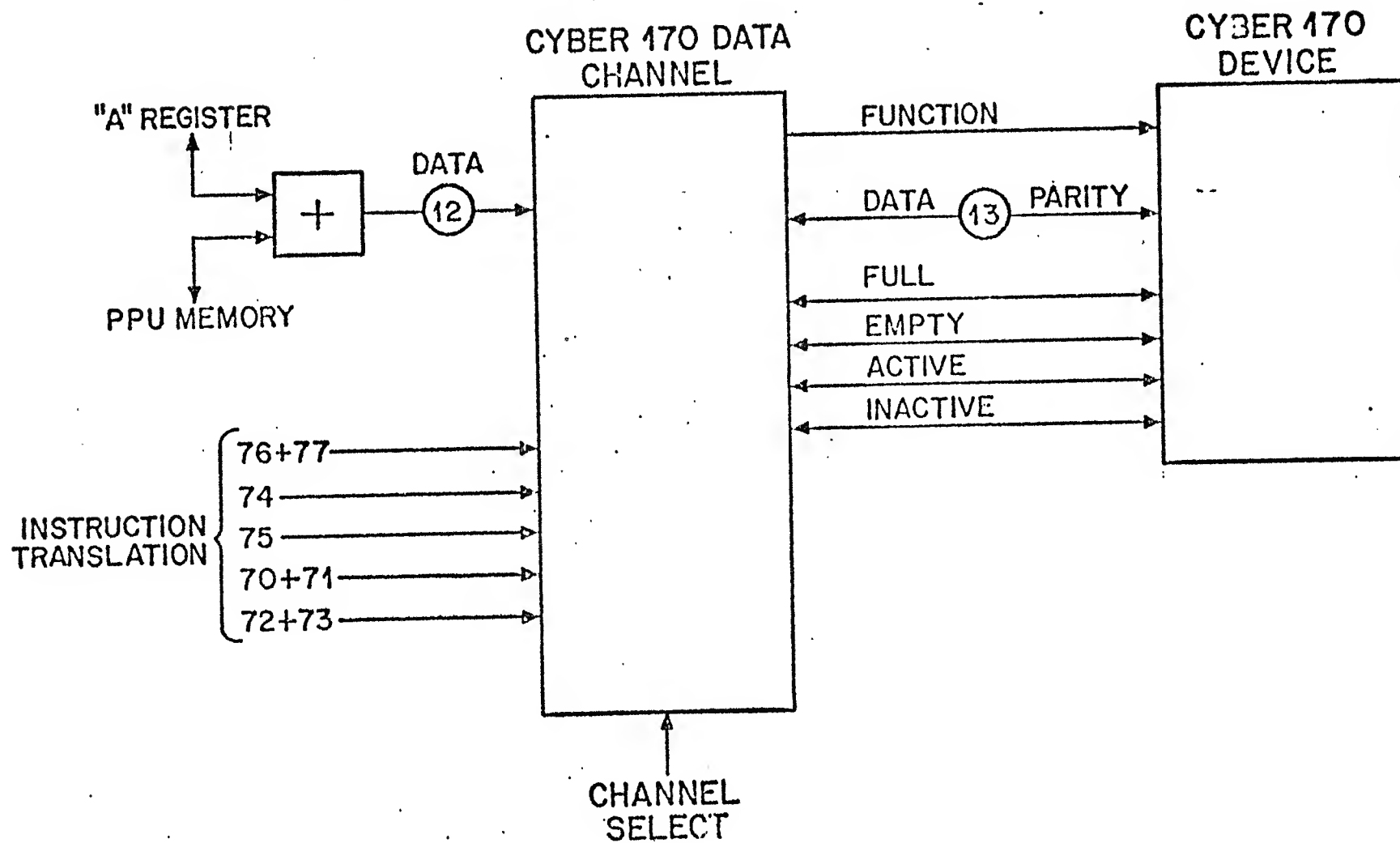


Figure 5-A-10. PPS Instructions 50 through 57

# CYBER 170 DATA CHANNEL



## DETAILED PAK DIAGRAM (PPS 3.12)

### CHANNEL CONTROL

PPS 3.12 shows hardware used primarily for channel control timing. Translations of I/O instructions and channel status are illustrated in PPS 3.11.

#### PPS CHANNEL SIGNALS AND STATUS

There is one AR module for each channel. The transmitter shown in the AR module (PPS 3.11) sends empty (EMPTY) and function (FUNY) signals to a selected channel if the conditions BWMT or BWFN, respectively, arise in the full/active control translator. Channels are selected by means of the select channel XX signal (SELXX). There is one such signal for each channel. The AP module's transmitter sends the deactivate signal (ACY - PPS 3.11) to the channels, but it originates in a similar way as the FUNY and EMPTY signals from the ARIA signal of the AR modules.

The transmission of these signals to channel X must be accompanied by a corresponding change in the empty (AQNFX) and inactive (AQNAX) states as indicated by flip-flops in the AQ modules (PPS 3.11). Signals ARMT (set empty), SACT (set active), and CACT (clear active), which can be generated by processor conditions or by responses from the channels, serve to make appropriate changes in channel conditions.

In the AQ modules there are several flip-flops working in conjunction to transmit full (FULLY), parity (PARY), and activate (AQAC) signals to the channels. In addition to storing channel conditions, the AQ module generates signals for transmission control within the processor. These are AQXM (enables transmission), AQSD (enables data entry into channel data in/out register), and AQCL (clears channel data in/out register before loading new data).

Note that signals are not necessarily sent when the corresponding change in channel condition occurs. For example, the function signal (FUNY) is sent rather than an active (ACTV) on a 76 or 77 instruction, although the active condition (AQNA) is set. Also, no signal is sent when condition changes are due to responses from the channels.

An example of a deactivate on a 75 instruction is shown in figure 4-2-5.

In this sequence, the signal ARIA enables transmission of the inactive signal IACY to the synchronizer. At the same time the CACT signal resets ACTV FF (T30), thereby setting the inactive flag AQNA. ACT FF (T1) is normally reset. It sets only momentarily on a 74 instruction to send the active signal to the synchronizer. All transmissions are gated by the tuned clock TXX which occurs at T30 time.

Note that BWIA, the deactivate GO signal, could arise on a minor cycle coincidental with either T10 or T30. If T30 occurs during BWIA, the send inactive FF is set directly and transmission occurs immediately on TXX. This is the most simple situation, illustrated in figure 4-2-5 by dotted lines where it differs from the case in which T10 is coincidental with BWIA. In the latter instance, when T30 does not occur during BWIA as shown by solid lines in figure 4-2-5, BWIA is held for one minor cycle in the inactive holding FF. This flip-flop, in turn, sets the send inactive flip-flop whose contents are transmitted on the channel by TXX.

For all such channel control signals, provision is made to retain channel commands until transmission time. In the AR paks, the empty, function and inactive holding FFs retain the transmit command until T30 time, while in AQ the full and active (T1) FFs retain their commands until transmission is completed.

The inactive indicator (AQNA) requires that either active FF be enabled to indicate an active state, whereas the transmit active signal (AQAC) specifically requires the ACTV flip-flop (T1) to enable a transmission. Thus on a 76 or 77 instruction, ACTV FF (T30) is set by the signal SACTOO to force an active state without sending the active signal to the synchronizer.

A similar arrangement exists for the full condition where both flip-flops (full FFs T1 and T30) must be set to transmit, but only one (T30) is set to disable the empty (AQNF) flag.

Figure 4-2-6 shows channel timing, including responses from the external device on a function instruction. The diagram shows that the channel becomes active at the same time as the function signal is transmitted. The channel responds with a deactivate signal. Timing restraints are noted at the bottom of figure 4-2-6.

#### PARITY

The parity holding flip-flop (PPS 3.11) outputs either a parity bit from the channel (CHPA) or a parity bit generated from data read from memory (CHWP). When CHPA is present, as an input from peripheral equipment, AVCHPD is generated and used in a parity check in the PPM channel parity checker. When the channel parity generator sends CHWP to the parity holding flip-flop on a 72 or 73 instruction, the selected parity transmitter sends this parity bit to the channel as PARY.

# PPS-0/PPS-1 STATUS DIFFERENCE

PPS-1 differs from PPS-0 in channel status assignments at dead start time. On a 20-PP system, all channels are forced active and empty at dead start time. However, in 14 or 17-PP systems, software requires that those channels assigned to nonexistent PPs must be forced inactive at dead start. Thus channels 04-11<sub>8</sub> and 07-11<sub>8</sub> of PPS-1 are forced inactive in 14 and 17-PP systems, respectively.

This is accomplished by using DSRT to force the CACT 03-09 (clear active) signals on all affected channels. The AR and AQ paks (PPS 3.12) carry out this alteration in normal status patterns.

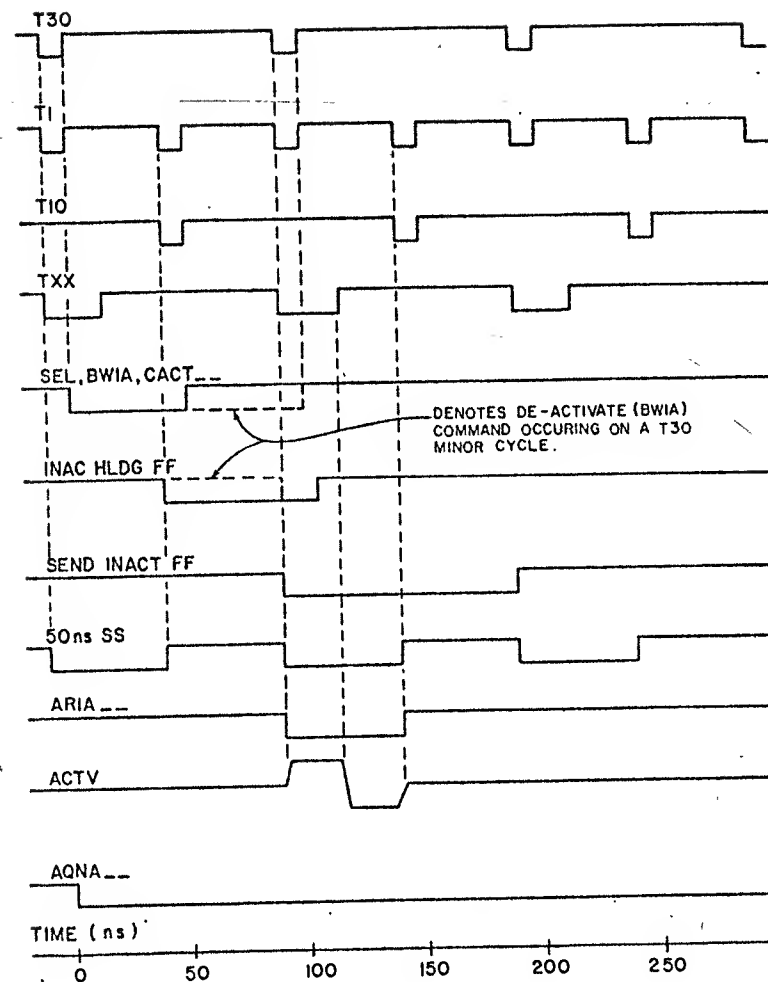
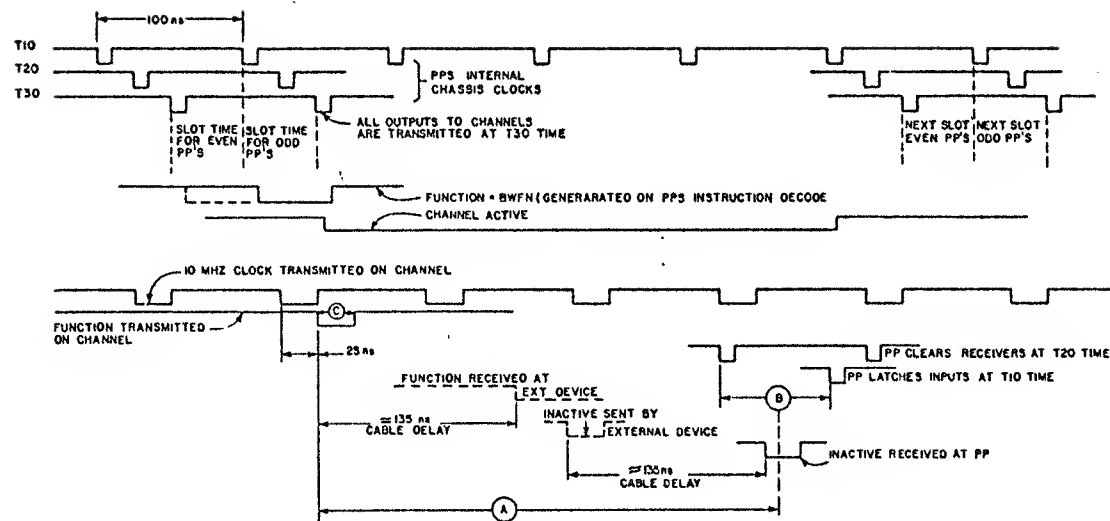


Figure 4-2-5. Deactivate Sequence



- NOTES • (A) TOTAL TURNAROUND TIME BETWEEN FUNCTION AND INACTIVE MEASURED AT THE PPS SHOULD BE  $310 \text{ ns} + 35 \text{ ns}$  TO MAINTAIN  $500 \text{ ns}$  CYCLE TIME
- (B) ALL INPUTS FROM THE CHANNEL TO THE PPS MUST ARRIVE AT THE PP IN THE  $75 \text{ ns}$  WINDOW INDICATED, TO AVOID LOST DATA (INPUTS MAY BE EARLIER OR LATER BY  $100 \text{ ns}$  MULTIPLES)
- (C) ALL AC TRANSMISSION PULSE WIDTHS ARE  $25 \text{ ns} + 5 \text{ ns}$  SIMILAR TIMING RELATIONSHIPS EXIST FOR DATA, FULL, EMPTY, ETC

Figure 4-2-6. I/O Timing for CYBER 170 I/O Channels

## DETAILED PAK DIAGRAM (PPS 3.13)

### READ PYRAMID

Control signals C5C4, C4C3, C3C2, C2C1 and C1Y from the BU module (PPS 3.13) gate the CM word into successive ranks of the read pyramid. At each rank, the 12 most significant bits are sent to the Y register.

The read pyramid receives data from CM via the CM data catching register. On trips 600 NF8 to 604 and on 613 to 617, a shift in the read pyramid is accomplished.

The CM data catching register actually forms C5. C5C4 gates bits 48-59 to the Y input selector (PPS 3.6). C4 receives the remaining 48 lower bits.

The BU and BK modules (PPS 3.13) together generate the control signals for the read pyramid.

On receipt of the CMDOL signal from CMC (which precedes the data by 50 ns), the data ready 2 flip-flop (DR2) sets. This enables the BKC5F signal, which in turn activates BUC5C4 when  $K = (6 \times 3)$  or when NF9. ( $K = 600$ ). On the next T1, data ready 1 (DR1) sets, thereby maintaining BKC5F = 1. DR1 resets on C5C4.T1 when a new CM word may be accepted.

Successive shifts in the read pyramid are dependent only on the instruction code which produces signals C4C3 to C1Y.

The signal TDR (PPS 3.13) = BKC5LA.T1. The AA module produces this signal unless either of the data ready FFs are set while the C5C4 flip-flop is reset. Thus, the CM data catching register holds a word from CM until it is shifted into C4 (indicated by BUC5C4).

#### TWO OUTSTANDING READ REQUESTS

If a data word remains unprocessed in the CM data catching register when a second CMDOL signal is received from CMC, DR2 sets. When the first word is shifted out of the CM data catching register on BUC5C4, the C5C4 flip-flop in the BK module sets and, through BKC5LA, gates the new word into the CM data catching register. At the same time DR2 shifts into DR1, making room for another word from CM.

There may be up to two PPs having outstanding requests for a CM read at any given time. The response from CMC may occur prior to the slot time of either of these PPs. To prevent the second PP from receiving the first's word, it is locked out by means of the BGC5EN signal which inhibits the generation of the BKC5F signal when the second PP is in the slot.

BKC5F is also inhibited by the C4C3 flip-flop. This prevents loss of data by ensuring that C4 is empty before C5 → C4.

#### PARITY

The CM read parity checker (PPS 3.13) produces a 17-ns SST 0000 signal when it detects a parity error on a CM read. This sets STC 000 (SC1T bit 0).

In the event that a single error is detected and corrected by means of the SECDED system, CMC sends the PPS a corrected data word. Normally the data is accompanied by an appropriate parity bit, but in this case the parity bit is inverted since it is generated from the data word before correction. To prevent the parity error signal that would be generated at the PPS in such cases, CMC also sends a bit (DPP01P) which inverts parity in the read pyramid receivers (KR pak - PPS 3.13).

---

## CHANNEL DESCRIPTION

The CYBER 170 series computer systems are configured with either 10 or 20 separate peripheral processors (PPs), any one of which can exchange data with external equipment connected to a data channel. A 10-PP system allows any processor access to any one of 12 data channels. A 20-PP system allows any of its processors access to any one of 24 data channels. Communication between a processor and an external equipment is via a synchronizer. Signal transfer between a data channel and a controller is by means of one-shot pulses that are converted, within the synchronizer, to the form required by the external equipment.

Each data channel has a 13-bit bidirectional register and several bidirectional control designators that define the status of the register and channel. Each data channel transfers 12-bit words and a parity bit at rates up to a maximum of one word every 500 nanoseconds when the PP is running at 2X speed. The one exception in which a 2-MHz transfer rate cannot be maintained occurs in a 20-PP system when a PP on one chassis is accessing a channel on the other chassis and, at the same time, a PP in the second chassis is executing channel instructions; e.g., PP-1 accessing channel 24<sub>8</sub> while PP-13<sub>8</sub> is also performing channel operations. In these exceptional cases, both PPs will operate at the 1-MHz rate.

Pulse communication is used on all data and control lines of a channel, all lines are synchronized to the PP clock system, and all channels may be in operation at the same time.

A "channel active" designator is set from an internal (PP) source, or from an external synchronizer, to reserve a channel for communication between a PP and a synchronizer (or another PP).

A "channel inactive" designator, originating either internally or externally, clears the "channel active" designator to terminate communication.

A "channel full" designator, set from an internal or external source, indicates that a 12-bit (plus parity) data word has been entered into the channel register.

A "channel empty" signal clears the "channel full" designator, which in turn clears the channel register.

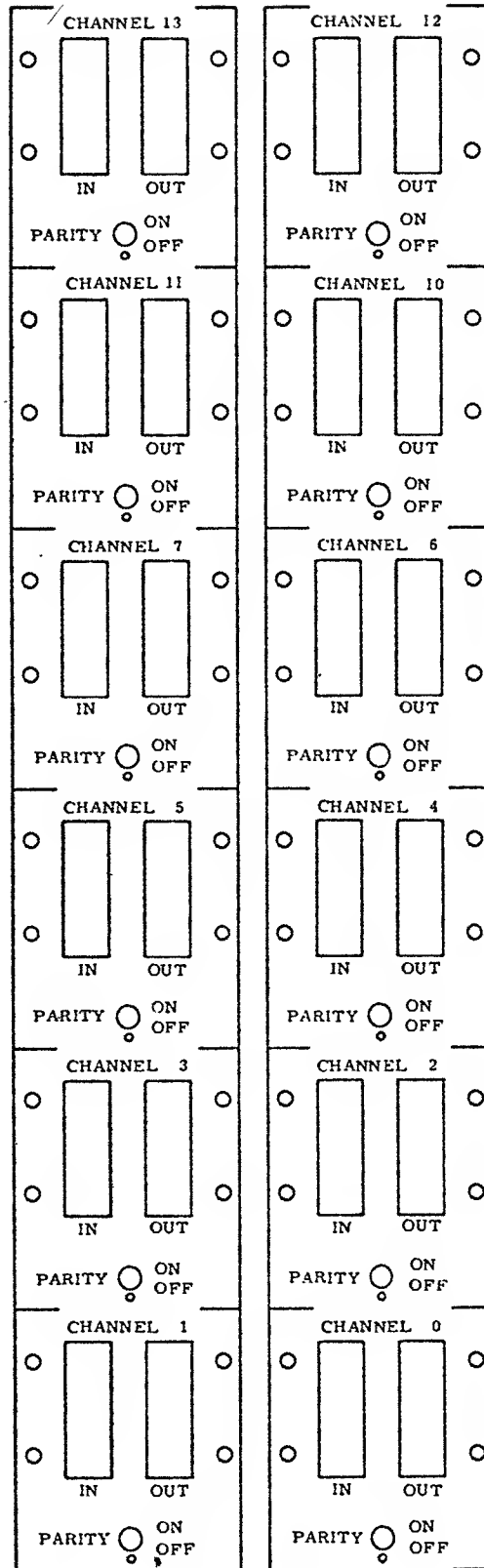


Figure 1-1. Data Channel Cable Connectors

The pulses on the data and control lines are one-shot, nonrepeat type transmissions, and all synchronizers must provide storage for the information. Control signals include a 10-MHz (100-nsec period) clock, a 1-MHz (1- $\mu$ sec period) clock, and a MASTER CLEAR for all external devices.

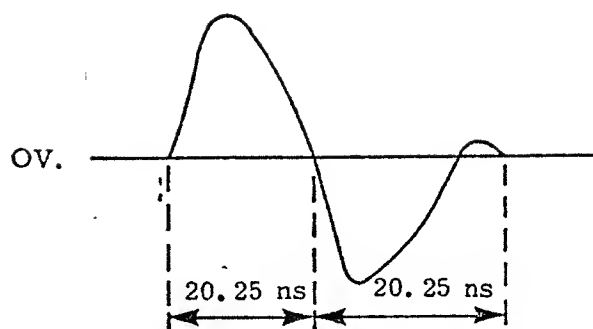
Channel data and control lines are grouped into two cables: input and output (refer to figure 1-1). The output cables carry processor signals to the synchronizers. The input cables carry synchronizer signals to the processor and the two clocks in the opposite direction. All devices on a channel connect to the data and control lines in a serial-parallel arrangement. Each synchronizer (with the exception of the data channel converter) samples the lines, and unconditionally relays all signals to the next-in-line synchronizer. Each synchronizer times the signal relay on the 10-MHz clock signal so that all are synchronous with the processor, but time-displaced from each other by one or more clock periods. The data channel converter (DCC) relays data only when not selected (the DCC does relay the function that selects it).

On a CYBER 170 with 10 PPs, channel 10<sub>g</sub> is permanently wired to the display controller (DSC) so that I/O channel data is available to external equipment only via the "pass-on" logic in the display controller. Thus, all I/O transfers on channel 10<sub>g</sub> are delayed by 100 nanoseconds. Channel 10<sub>g</sub> is wired as a normal channel on the second chassis of a 20-PP system when a second display controller is not installed.

## **SIGNAL SPECIFICATIONS**

The data channel signal specifications are listed in table 1-1. Separate input and output cables are used for each data channel. Each cable line originates and terminates on the PPS logic chassis. Cable lines have a fixed length of 22.9 meters (75 feet), including the lengths internal to both the PPS and external synchronizer.

A logical "1" signal, measured at the circuit terminals of the sending device, is illustrated in figure 1-2 and described in table 1-2. No signal is impressed on the line for a logical zero. Input circuits in the external equipment must terminate the line in its characteristic impedance.



MAX VOLTAGE SWING =  $\pm 2.7V$  p-p

MIN VOLTAGE SWING =  $\pm 2.1V$  p-p

Figure 1-2. Output Pulse Characteristics

Measured at output pin of TR. looking into a  $75\Omega$  impedance.

TABLE 1-1. DATA CHANNEL COAXIAL CABLE LINES

Input Cable	PIN	Color Code	Output Cable
Data bit 0	A	90	Data bit 0
Data bit 1	B	91	Data bit 1
Data bit 2	C	92	Data bit 2
Data bit 3	D	93	Data bit 3
Data bit 4	E	94	Data bit 4
Data bit 5	F	95	Data bit 5
Data bit 6	H	96	Data bit 6
Data bit 7	J	97	Data bit 7
Data bit 8	K	98	Data bit 8
Data bit 9	L	99	Data bit 9
Data bit 10	M	900	Data bit 10
Data bit 11	N	901	Data bit 11
Active	P	902	Active
Inactive	R	903	Inactive
Full	S	904	Full
Empty	T	905	Empty
Clock (10 MHz)	U	906	Function
Clock (1 MHz)	V	907	Master Clear
Input Data Parity	W	908	Output Data Parity

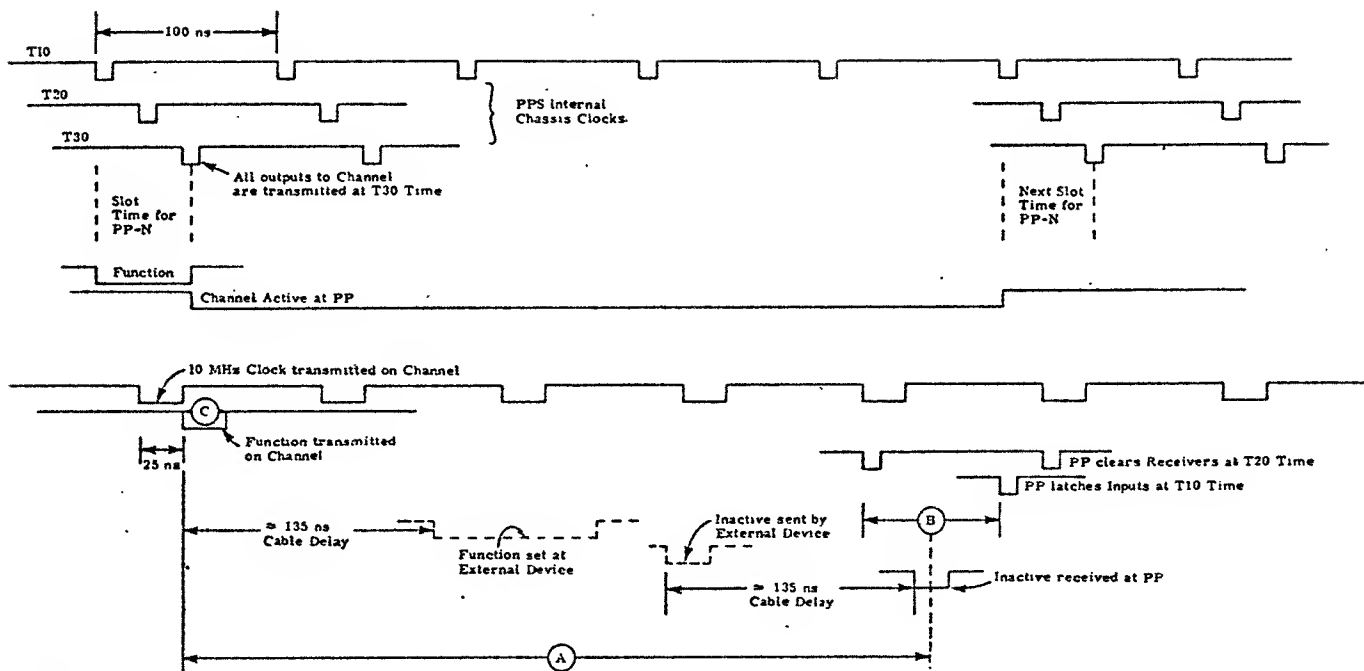
TABLE 1-2. LINE CHARACTERISTICS

Parameter	Description
Line length (maximum)	22.9 meters (75 feet) typical CDC cable
Pulse amplitude at output of the transmitter	2.3 volt peak at 32 milliamperes into a 70-73 ohm coaxial cable - the cable to be terminated in its approximate characteristic impedance
Rise time	} at the output of the transmitter 2 nanoseconds
Fall time	
Line capacitance	70.5 picofarads/meter (21.5/foot) maximum (typical CDC cable)
Line attenuation	0.15 decibels/meter (0.045/foot) (typical)
Voltage rating	30 volts maximum

## NOTE

22.9 meters (75 feet) line length is the total length from the transmitter to the receiver. This includes the short cables on the PP and synchronizer chassis.

## SIGNAL TIMING



### Notes:

- (A) Total Turnaround Time between Function and Inactive measured at the PPS should be  $410 \text{ ns} \pm 35 \text{ ns}$  to maintain  $500 \text{ ns}$  Cycle Time.
- (B) All inputs from the Channel to the PPS must arrive at the PP in the  $75 \text{ ns}$  window indicated, to avoid lost data (inputs may be earlier or later by  $100 \text{ ns}$  multiples).
- (C) All AC Transmission Pulse Widths are  $25 \text{ ns} \pm 5 \text{ ns}$ .

Similar timing relationships exist for Data, Full, Empty, etc.

The MASTER CLEAR is pulsed at  $4096\text{-}\mu\text{sec}$  periods for as long as the DEADSTART switch (on the deadstart panel) is in the ON position, or while the DEADSTART button (on the display console) is pressed. Each deadstart cycle causes a train of pulses to be transmitted, on all channels, for a  $1\text{-}\mu\text{sec}$  period.

## SIGNAL RELAY

Signal relay is necessary in all synchronizers on a channel except in the one at the end of the line. Signal timing through the relay is at a 10-MHz rate. Each synchronizer appears as the processor to the next-in-line synchronizer except for a time lag. This time lag depends upon the cable length and the internal circuiting of each synchronizer between it and the data channel. Each synchronizer samples and stores all output signal lines in addition to sending them to the next-in-line synchronizer.

Each synchronizer also transfers all input signal lines to the next-in-line synchronizer. In addition, each synchronizer must provide for entering its signals onto the same lines. Therefore, the network feeding a processor (or the next-in-line synchronizer) is an OR combination of the synchronizer and the one feeding it.

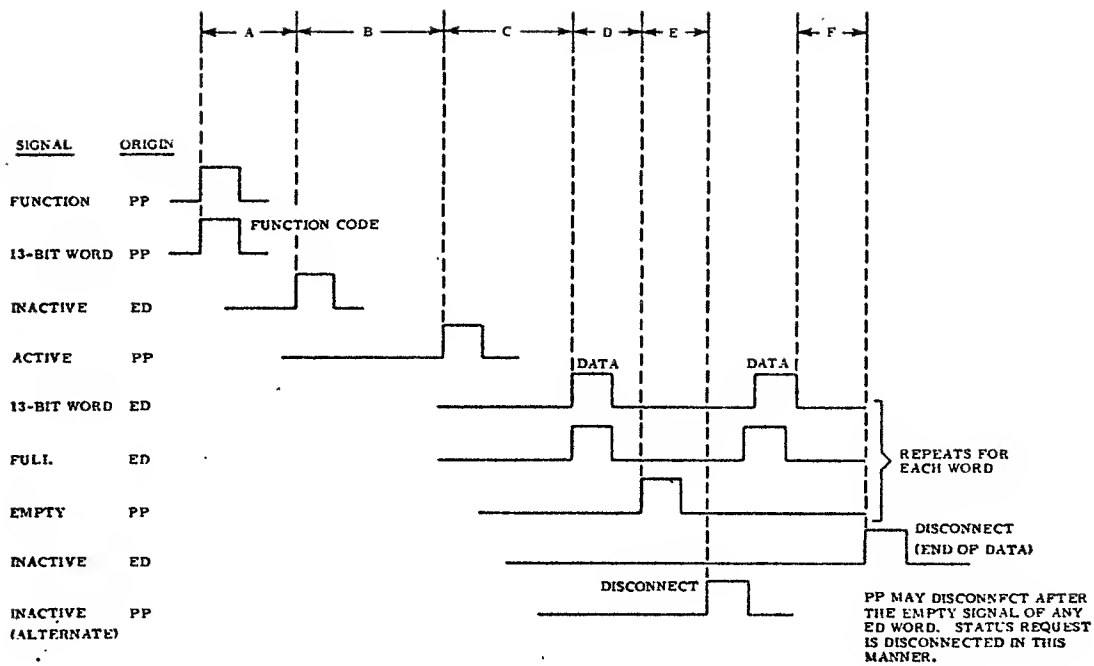
Computation of cable time and signal delay time through internal hardware allows a synchronizer to place the signals on the lines, for relay to the next-in-line synchronizer, at a time relative to the processor delayed by an integral multiple of 10-MHz clock periods. Thus, each synchronizer appears as the processor to the next-in-line synchronizer.

## TYPICAL DATA INPUT SEQUENCE

An external device sends data to the processor by way of a synchronizer in the following manner (refer to figure 2-1):

1. The processor places a function word in the channel register and sets the "full" and "channel active" flags. Coincidentally, it sends the function word and a "function" signal to all synchronizers. The "function" signal tells all synchronizers to sample the word, and identifies it as a function code (as opposed to a data word). The code selects a synchronizer and a mode of operation. Non-selected synchronizers clear, and only the selected synchronizer is turned on. A synchronizer will not select if it detects a parity error in the function word.
2. The synchronizer sends an "inactive" signal to the processor, indicating acceptance of the function code. This signal drops the "channel active" flag which, in turn, drops the "full" flag and clears the channel register.

3. The processor sets the "channel active" flag and sends an "active" signal to the synchronizer signalling the device to start sending data.
4. The device reads a word and then sends that word to the channel register with a "full" signal that sets the "channel full" flag coincident with the data arrival.
5. The processor stores the word; drops the "full" flag and returns an "empty" signal indicating acceptance of the word. The device clears its data register and prepares to send the next word. The detection of a parity error by the channel causes a signal to be sent to the status and control register, indicating this condition and the channel on which it occurred.
6. Steps 4 and 5 are repeated for each word transferred.
7. At the end of the transfer, the synchronizer clears its "active" condition and sends an "inactive" signal to the processor to indicate "end-of-data". The signal clears the "channel active" flag, disconnecting the synchronizer and the processor from the channel.
8. As an alternative, the processor may choose to deactivate the channel before the device has sent all its data. The processor does this by dropping the "active" flag and sending an "inactive" signal to the synchronizer that immediately clears its active condition and sends no more data, although the device may continue to the end of its record or cycle (e.g., a magnetic tape unit continues to "end-of-record" and stops in the record gap).



PP = peripheral and control processor; ED = external device

- A. Time is a function of ED  
PP recognizes inactive 1 major cycle time\* (or an integral multiple of major cycle times) after function. Inactive must have been received by the PP sometime before this.
- B. Time is a function of PP  
Minimum time is 1 minor cycle time\*; actual time is a function of the PP program.
- C. Time is a function of ED
- D. Time is a function of PP  
Minimum time is 1 minor cycle time; maximum time is an integral multiple of up to 9 minor cycle times (to allow operation within 1 major cycle time).
- E. Time is a function of PP.  
Minimum time is 3 major cycle times; maximum time is an integral multiple of major cycle times.
- F. Time is a function of ED

\* Major cycle time is 500 nanoseconds or 1 microsecond (500 nsec or 1.0 $\mu$ s modes of operation).  
Minor cycle time is 50 nanoseconds or 100 nanoseconds (500 nsec or 1.0 $\mu$ s modes of operation).

Figure 2-1. Data Input Sequence, Data Channel

## STATUS REQUEST

A status request is a special one-word data input transfer in which an external device indicates a "ready" or "error" condition to a processor (refer to figure 2-1).

1. The processor places a function word in the channel register and sets the "full" and "channel active" flags. At the same time, it sends the word and a "function" signal to all synchronizers. The "function" signal tells all synchronizers to sample the word as a function code (as opposed to a data word). The code selects a synchronizer and places it in the "status" mode. Nonselected synchronizers clear, and only the selected synchronizer is turned on. A synchronizer will not select if it detects a parity error.
2. The synchronizer sends an "inactive" signal to the processor indicating acceptance of the status function code. The signal drops the "channel active" flag which, in turn, drops the "full" flag and clears the channel register.
3. The processor sets the "channel active" flag and sends an "active" signal to the synchronizer which signals the device to send the status word.
4. The synchronizer sends the status word, drops the "full" flag and returns an "empty" signal, indicating acceptance of the word. The detection of a parity error by the processor causes a signal to be sent to the status and control register, indicating this condition and the channel on which it occurred.
5. The synchronizer accepts the word and sends an "empty" signal to the processor where it clears the channel register and drops the "full" flag.
6. The processor drops the "channel active" flag and sends an "inactive" signal to turn the synchronizer off.

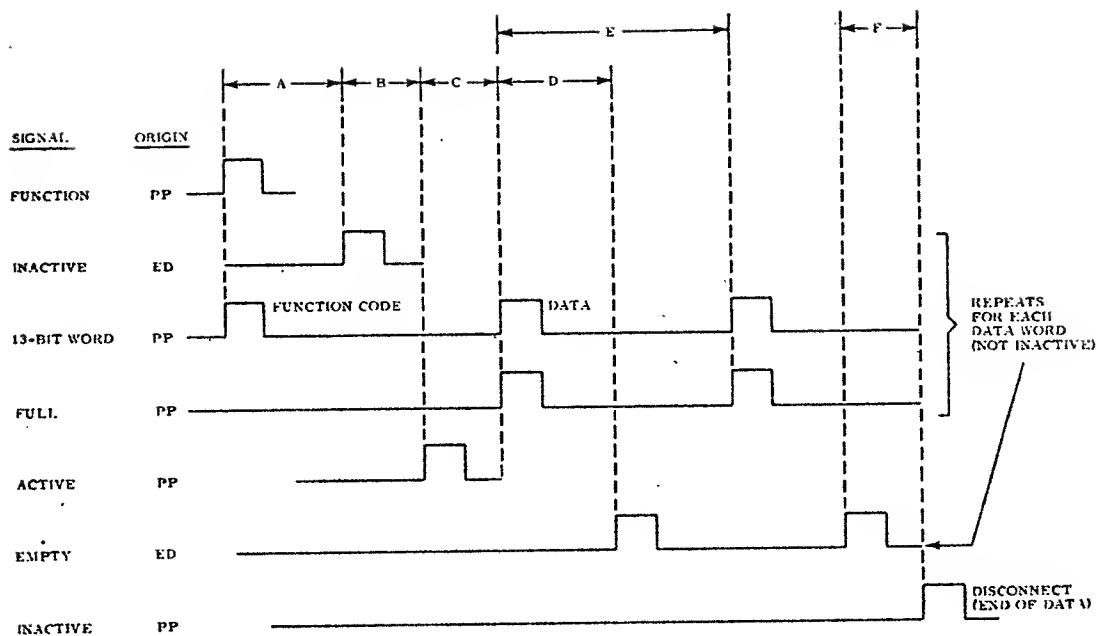
## TYPICAL DATA OUTPUT SEQUENCE

The processor sends data to an external device in the following manner (refer to figure 2-2).

1. The processor places a function word in the channel register and sets the "full" and "channel active" flags. Coincidentally, it sends the word and a "function" signal to all synchronizers. The "function" signal informs all synchronizers to sample the word and identifies it as a function code. The code selects a synchronizer and a mode of operation. Nonselected synchronizers clear, and only the selected synchronizer is turned on. A synchronizer will not select if it detects a parity error in the function word.
2. The synchronizer sends an "inactive" signal to the processor, indicating acceptance of the function code. The signal drops the "channel active" flag which, in turn, drops the "full" flag and clears the channel register.
3. The processor sets the "channel active" flag and sends an "active" signal to the synchronizer which signals to the device that data flow is starting.
4. The processor places a data word in the channel register and sets the "full" flag. Coincidentally, it sends the word and a "full" signal to the synchronizer.
5. The synchronizer accepts the word and sends an "empty" signal to the processor, where it clears the channel register and drops the "full" flag.
6. Steps 4 and 5 are repeated for each synchronizer word.
7. After the last word has been transferred and acknowledged by the synchronizer, the processor drops the "channel active" flag and sends an "inactive" signal to the synchronizer, turning it off.

## TREATMENT OF PARITY ERROR

One odd parity bit is added to all the 12-bit words (data or function) prior to loading the channel output registers. Thus, a 13-bit word is transferred on the output channel. Likewise, a 13-bit word is expected on input transfers from the external devices. The incoming 13-bit words are held in the appropriate "channel data-in" registers, where the odd parity is checked and only the 12 data bits are transferred to the appropriate register or registers. Parity bits are transferred over the "W" wire in both input and output coaxial cables.



PP = peripheral and control processor; ED = external device

- |                             |   |
|-----------------------------|---|
| A. Time is a function of ED | PP recognizes inactive 1 major cycle time* (or an integral multiple of major cycle times) after function.   |
| B. Time is a function of PP | Minimum time is 1 minor cycle time*; actual time is a function of the PP program.   |
| C. Time is a function of PP | Minimum time is 2 or 4 major cycle times depending on instruction; actual time is a function of the PP program.                                       |
| D. Time is a function of ED | Minimum time is 1 minor cycle time; maximum time is an integral multiple of up to 9 minor cycle times (to allow operation within 1 major cycle time). |
| E. Time is a function of ED | Minimum PP time is 1 major cycle time.  |
| F. Time is a function of PP | Minimum time is 2 major cycle times after empty from ED.  |

\* Major cycle time is 500 nanoseconds or 1 microsecond (500 nsec or 1.0 $\mu$ s modes of operation).  
 Minor cycle time is 50 nanoseconds or 100 nanoseconds (500 nsec or 1.0 $\mu$ s modes of operation).

Figure 2-2. Data Output Sequence, Data Channel

There is no parity on the control signals transferred to and from the peripheral processors on the channels.

In the event of a parity error during an input data transfer on any of the channels of a PPS, a parity error indication is sent to the status and control register. Bits from 024 to 035 are set for the respective channel parity errors of the 12 channels. The PPS continues in its normal mode of operation, and any corrective action to retrieve the system is left to the system's software. If a transmission parity error occurs during an output transfer, it is expected that the external device synchronizers will set an error bit in their respective error registers. Any other action taken may vary from device to device depending upon its capabilities.

The internal data channel converter (part of PPS-0) will respond as follows when it detects a parity error on data transfer from the channel:

1. Parity Error in Function Codes

- (a) The "connect" and "function" signals to the 3000 series equipment are blocked.
- (b) The converter does not send an "inactive" signal to the channel.
- (c) The parity error status bits 2 and 11 in the converter are not set.
- (d) The function is not executed.

2. Parity Error on Data Words (including functional data on Mode II Connect or Function)

- (a) Both parity error status bits 2 and 11 are set in the converter status word.
- (b) The data is used as normal. The parity bit received from channel is sent unchanged to the 3000 series synchronizers. The connected synchronizer will also detect the parity error and react appropriately.

To Mode II connect or function code parity errors, the DCC will generate an "empty" signal after a delay of 100 nanoseconds.

3. Parity Error on Data Received by the DCC from the 3000 Series equipment

- (a) Bit 2 is set in the DCC status word.
- (b) The data is sent to the channel with uncorrected parity.

#### 4. Parity on Status Information

No parity is generated on status information transferred from the 3000 series equipment to the DCC. The DCC generates and transmits parity for the status information to the channel.

For external devices with no parity capability, there are 12 switches on the PPS chassis to disable parity checking on data received on any of the 12 channels.

### TEST OF PARITY SYSTEM

Bit 120<sub>8</sub> of the status and control register (SCR) allows testing of the parity network in the PPS. When set, this bit forces zero parity transfer on all 12 channels; i.e., this function cannot be selected for a single channel.

## TYPICAL CONNECTIONS

Synchronizers are connected to an associated data channel as shown in the typical configuration in figure 3-1.

Several synchronizers may be connected to a common data channel provided none have identical select function codes.

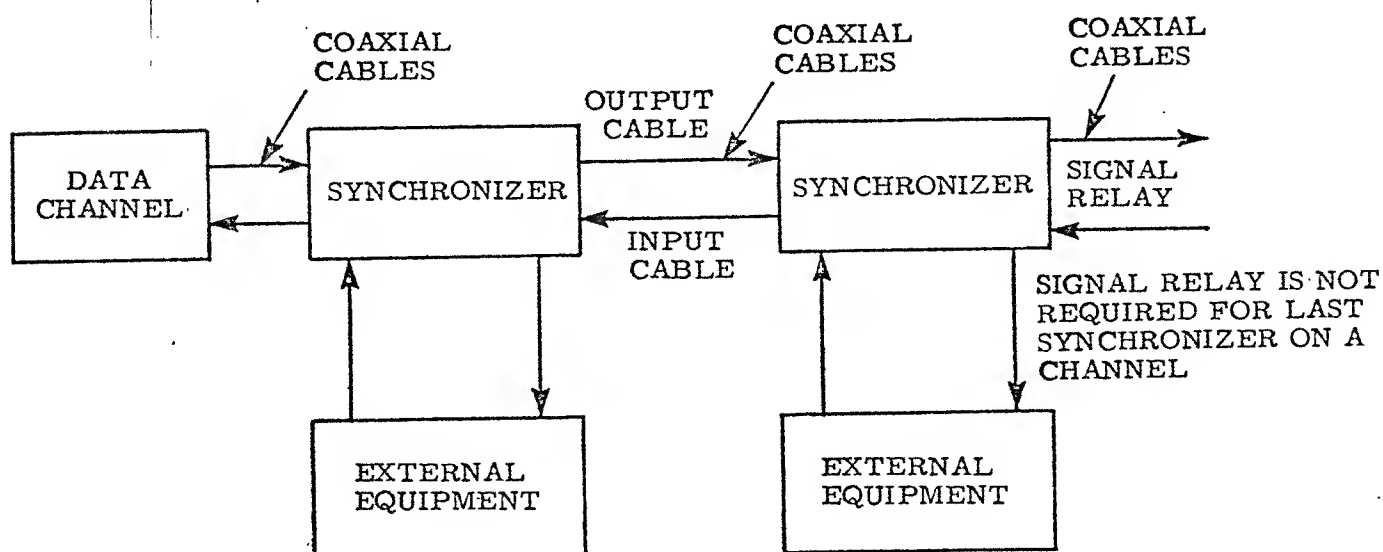


Figure 3-1. Typical Synchronizer/Equipment Connections

## SPECIAL CONNECTIONS

Two special equipments have a modified configuration. These are the DCC (data channel converter) and the 6682/83 satellite coupler (refer to figures 3-2 and 3-3).

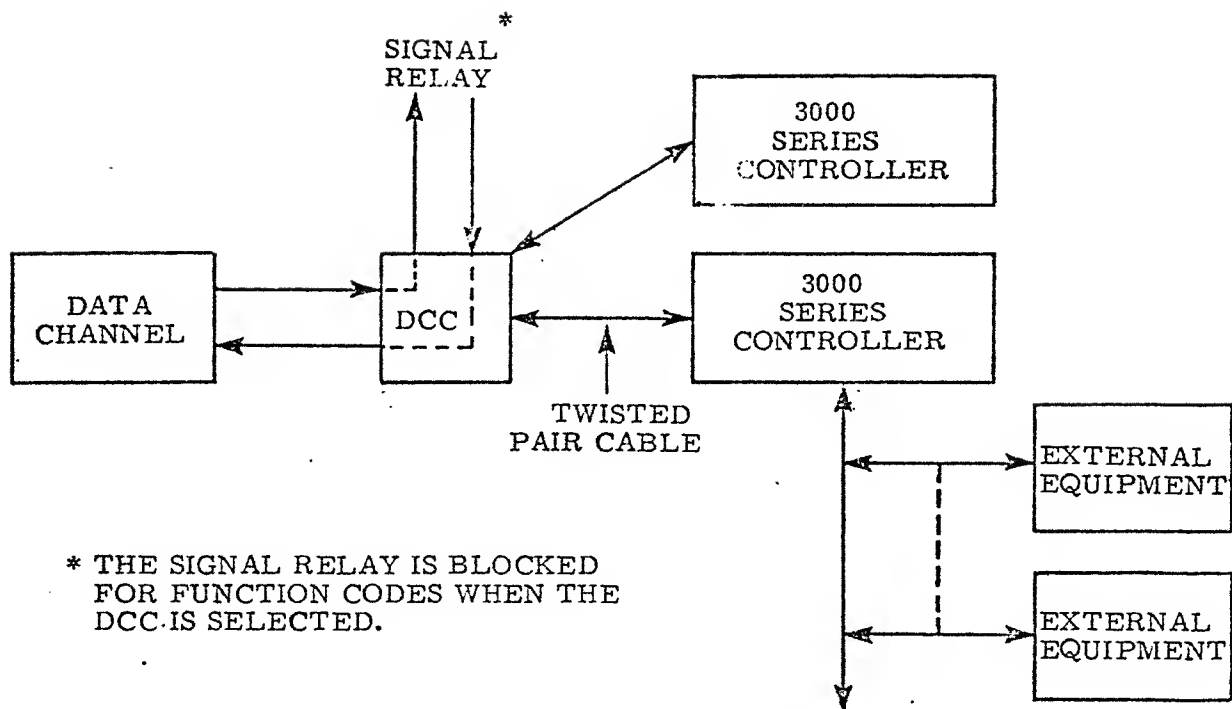


Figure 3-2. Typical DCC Configuration

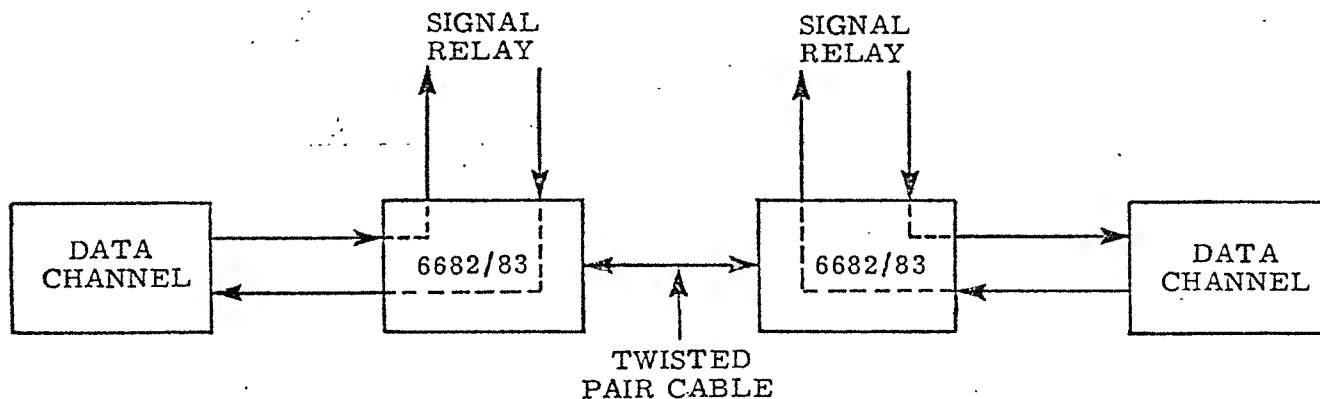


Figure 3-3. Typical 6682/83 Configuration

The 6681 may be selected by a select function code or by performing a master clear.  
 The 6682/6683 and all other controllers may be selected by a select function code only.

CODE	
AJM 64	JUMP TO M IF INT OR EXT CHAN D ACTIVE
IJM 65	JUMP TO M IF INT OR EXT CHAN D INACTIVE
IJM 66	JUMP TO M IF INT OR EXT CHAN D FULL
EJM 67	JUMP TO M IF INT OR EXT CHAN D EMPTY

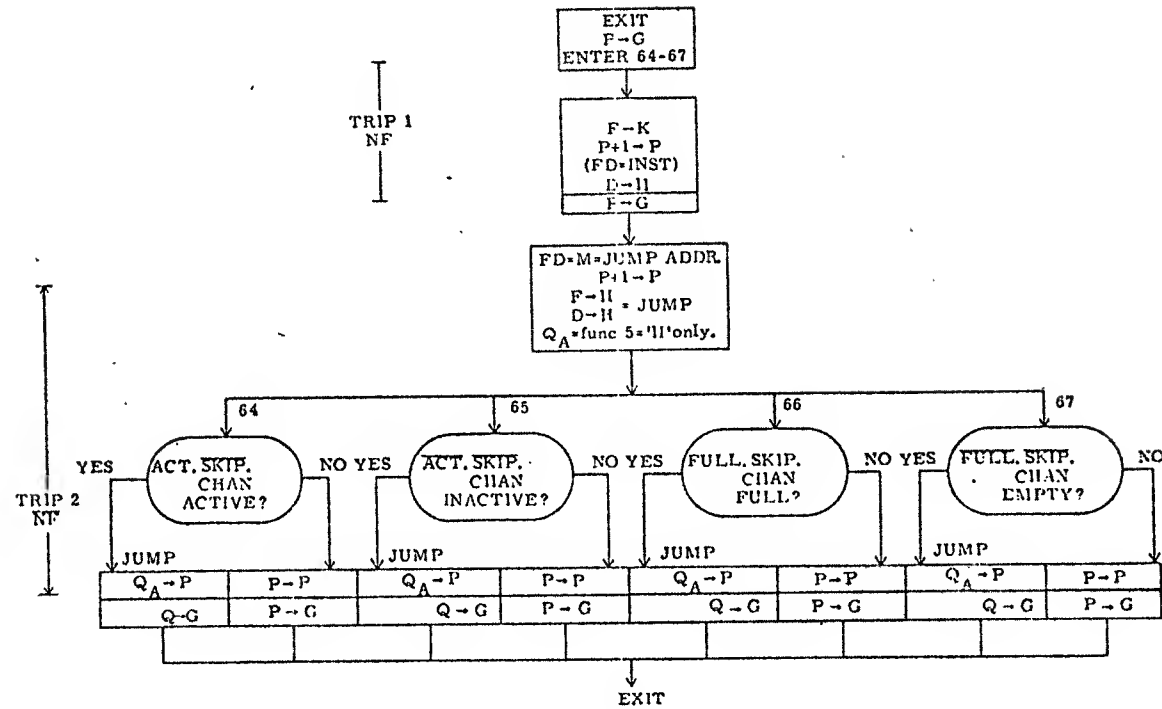


Figure 5-A-15. PPS Instructions 64 through 67

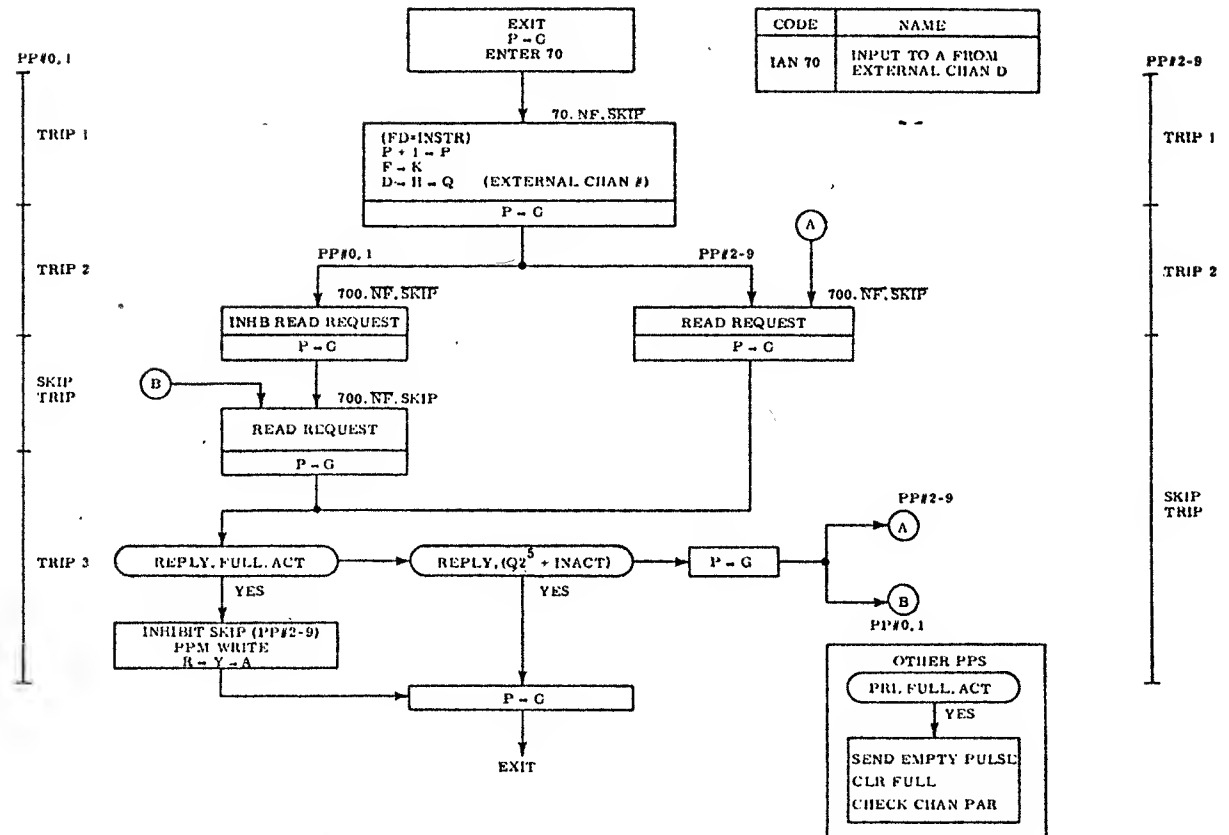
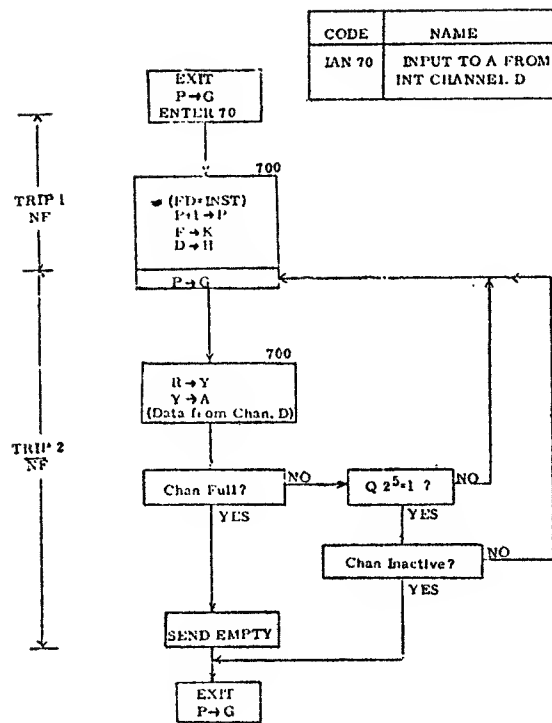


Figure 5-A-16. PPS Instruction 70

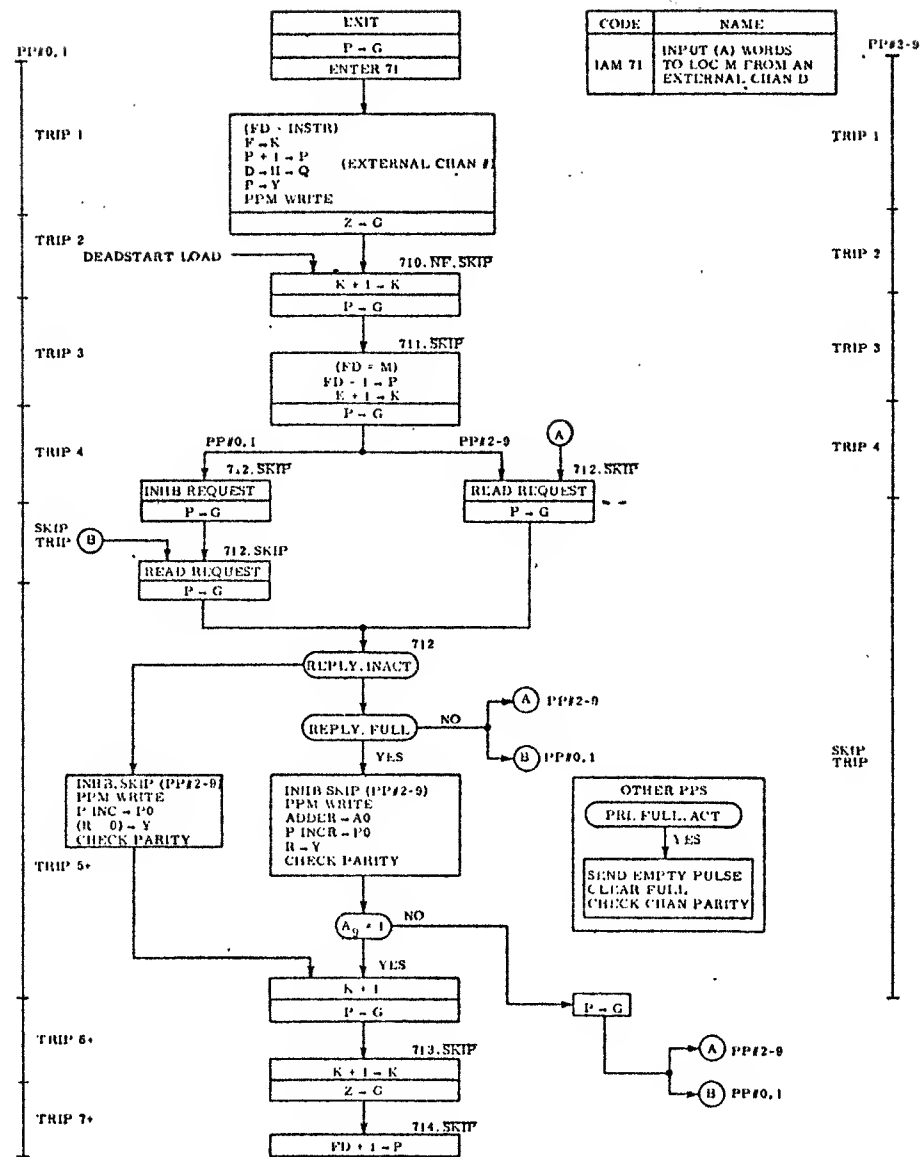
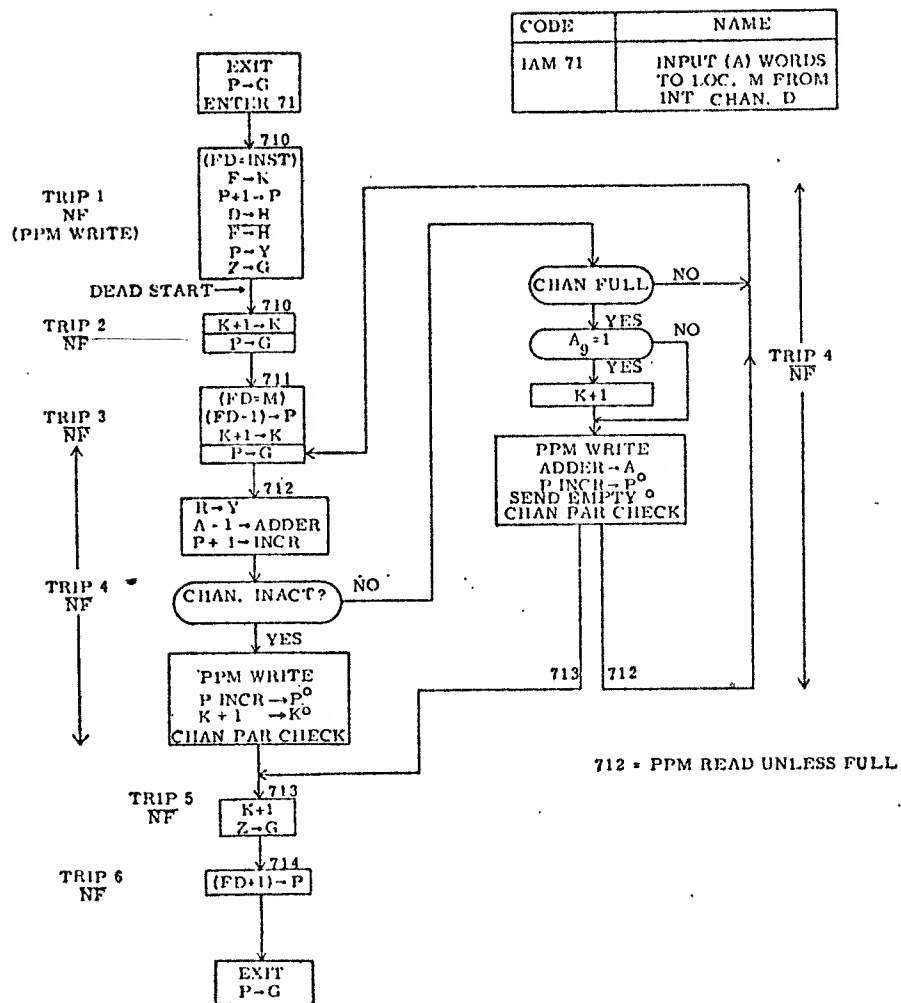


Figure 5-A-17. PPS Instruction 71

CODE	NAME
OAM 73	OUTPUT (A) WORDS FROM LOC M ON INT CHAN D

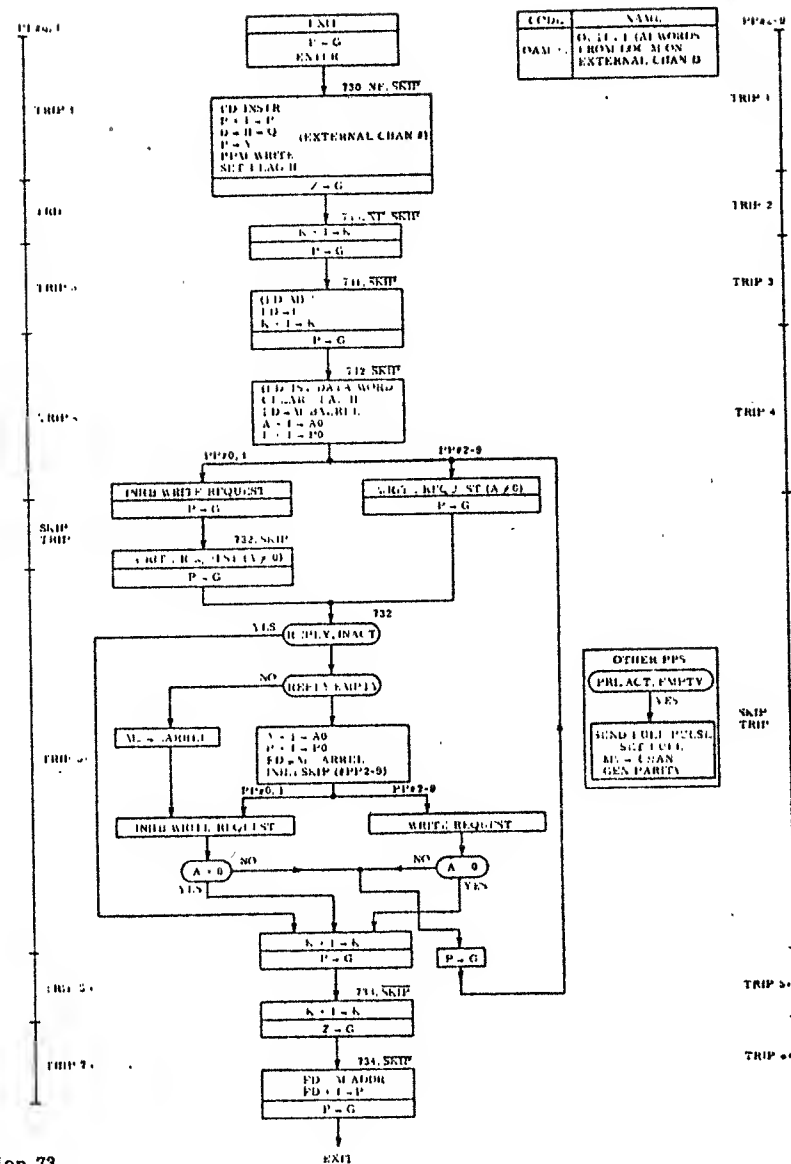
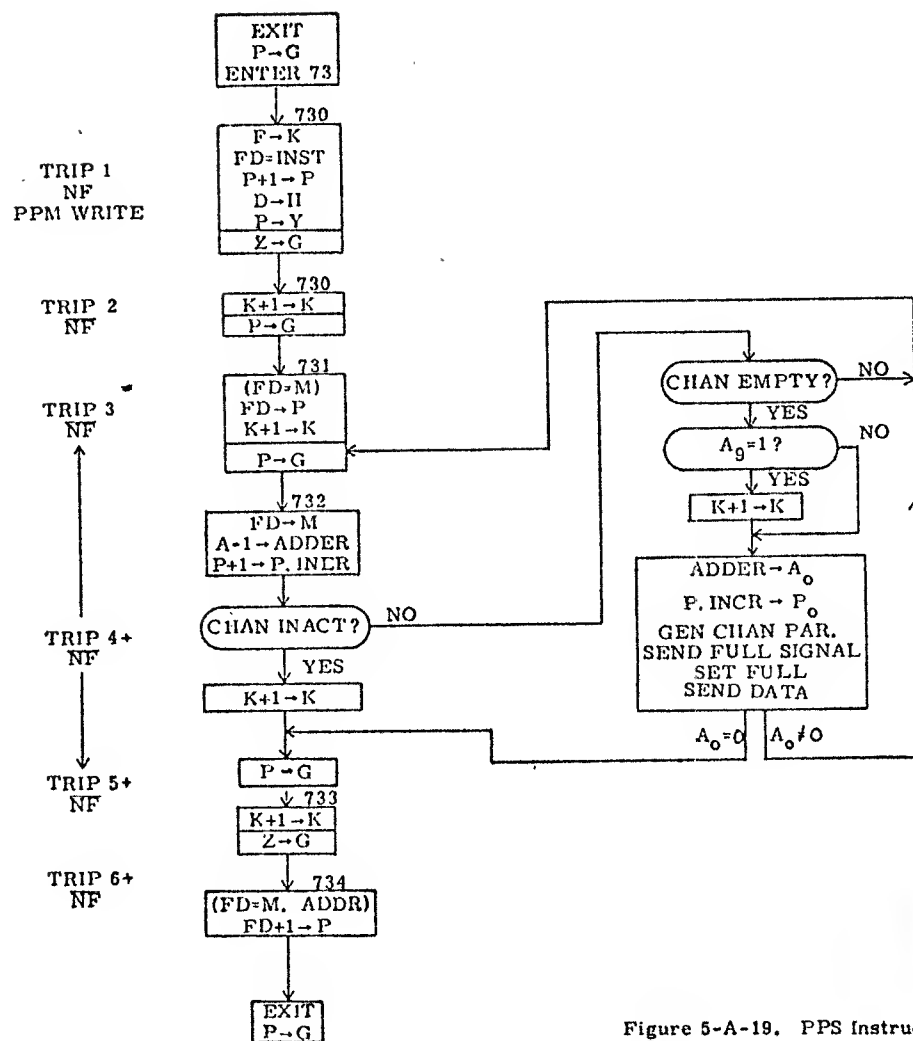


Figure 5-A-19. PPS Instruction 73

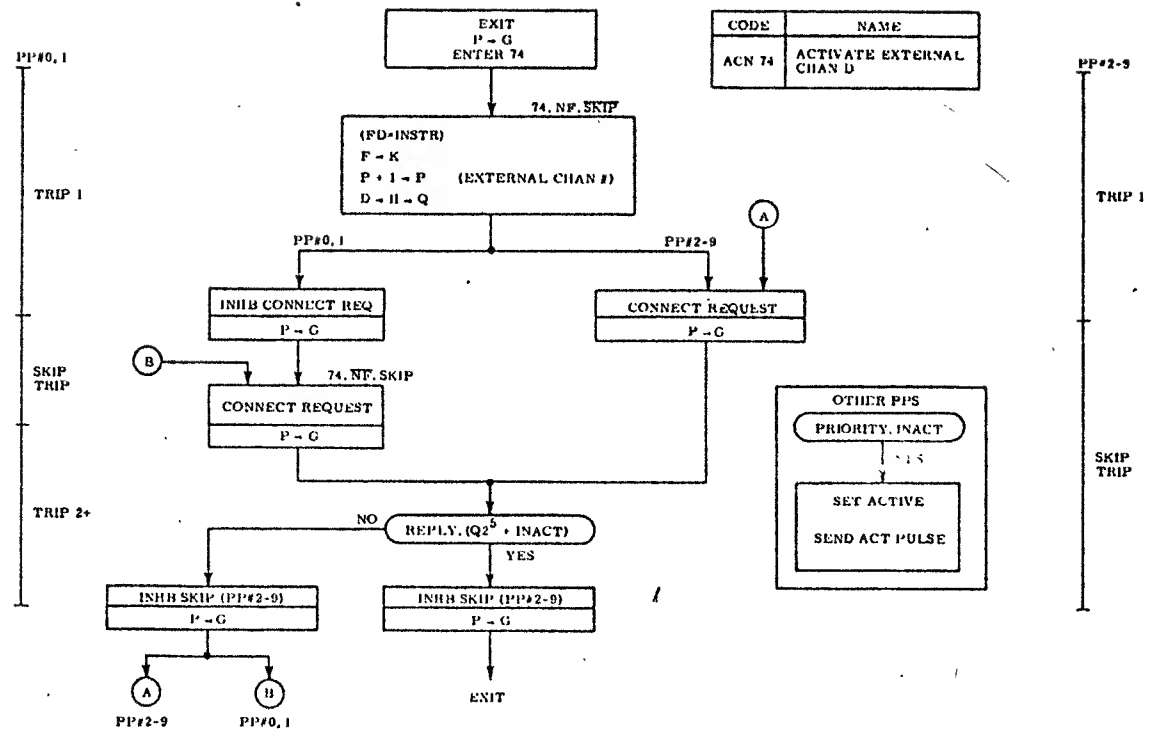
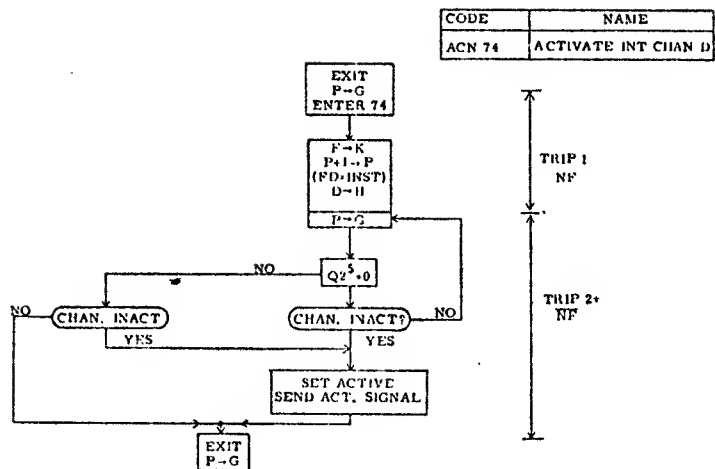
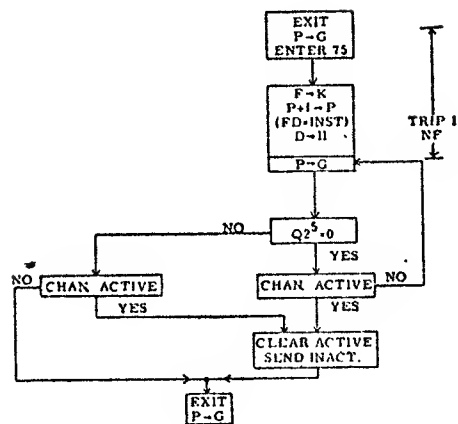


Figure 5-A-20. PPS Instruction 74

CODE	NAME
DCN 75	DISCONNECT INT CHANNEL D



TRIP 2  
NF

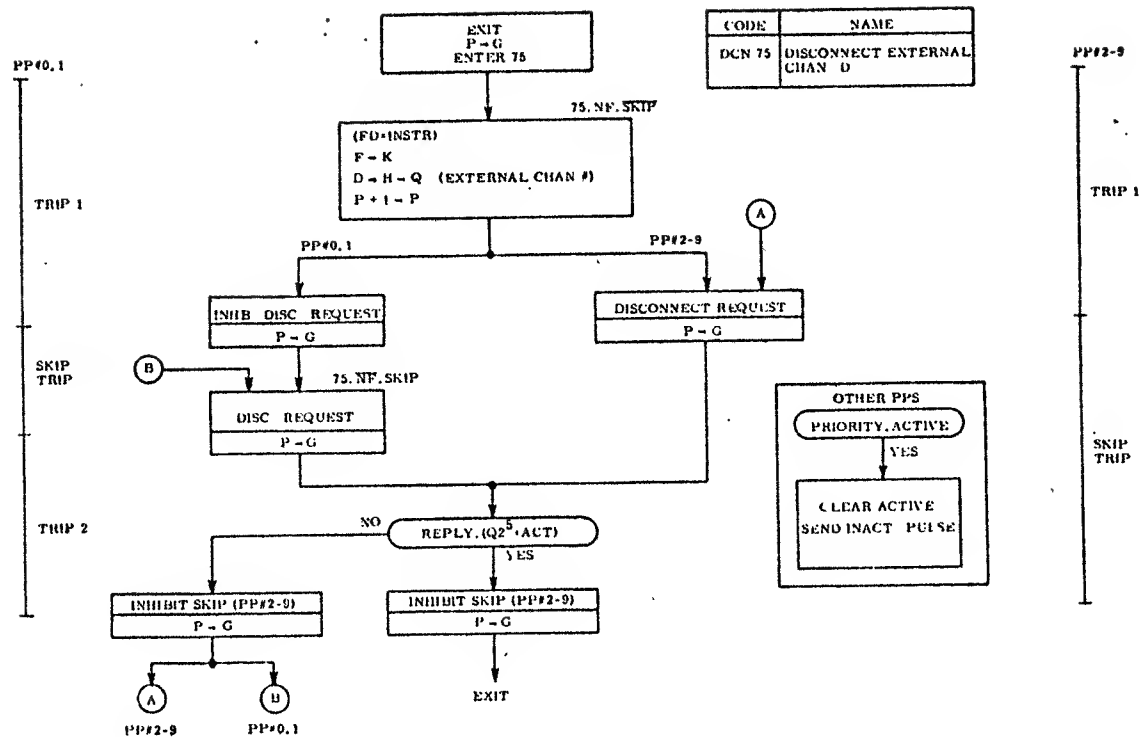


Figure 5-A-21. PPS Instruction 75

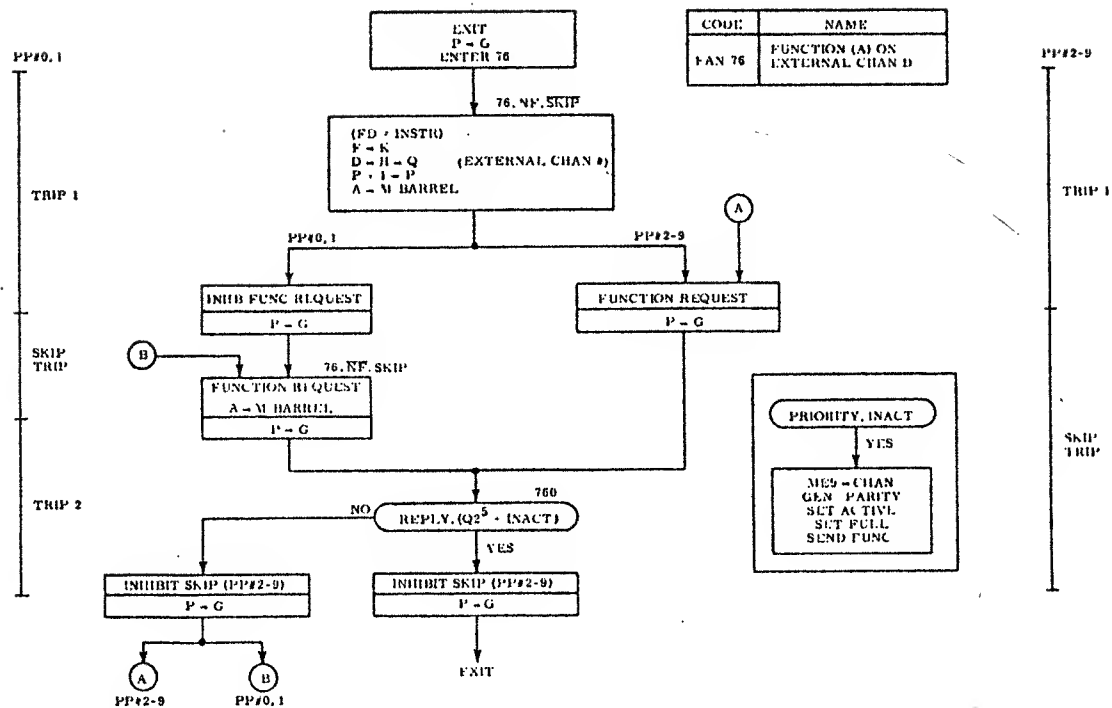
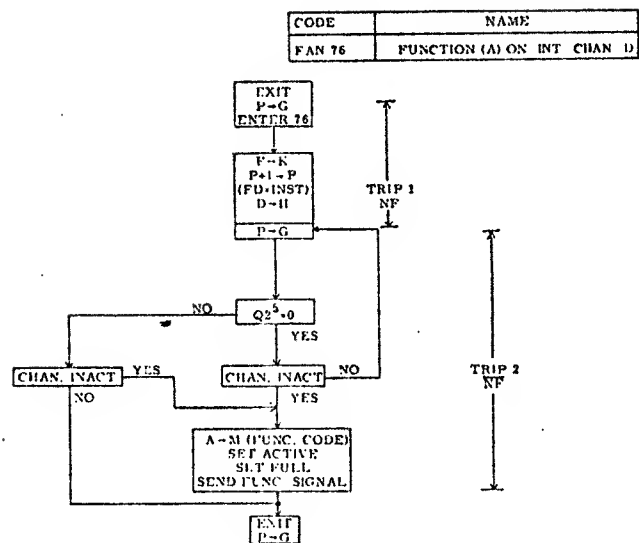


Figure 5-A-22. PPS Instruction 76

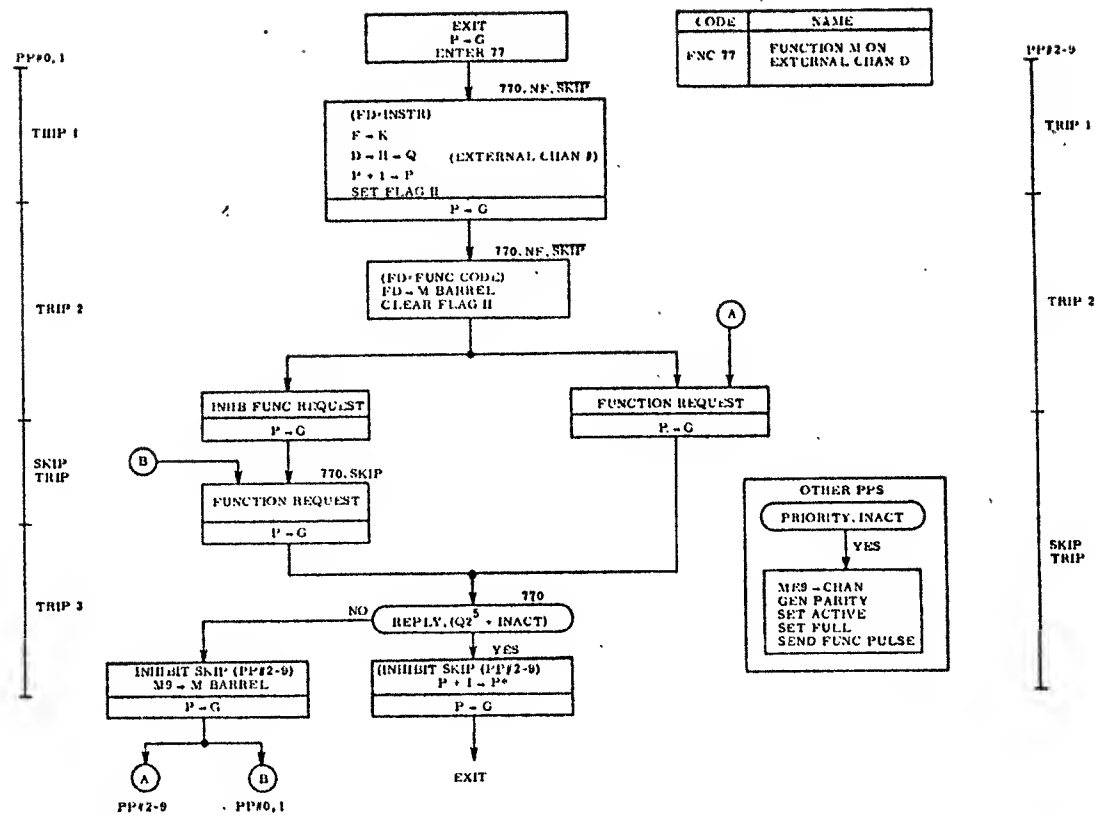
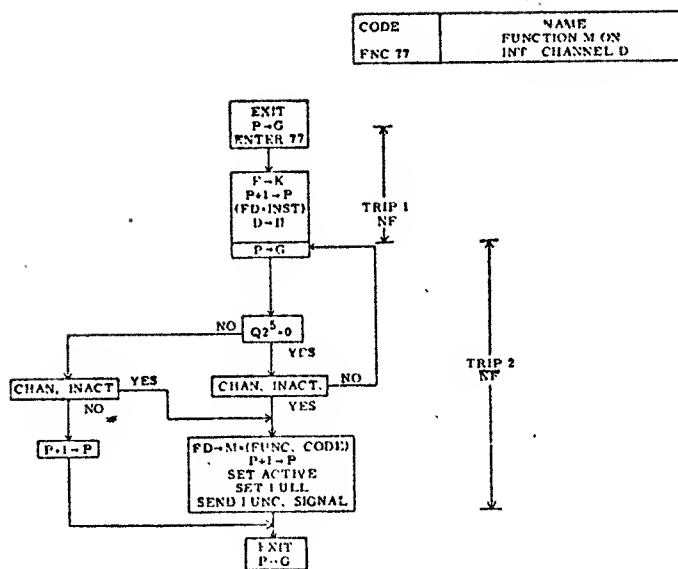
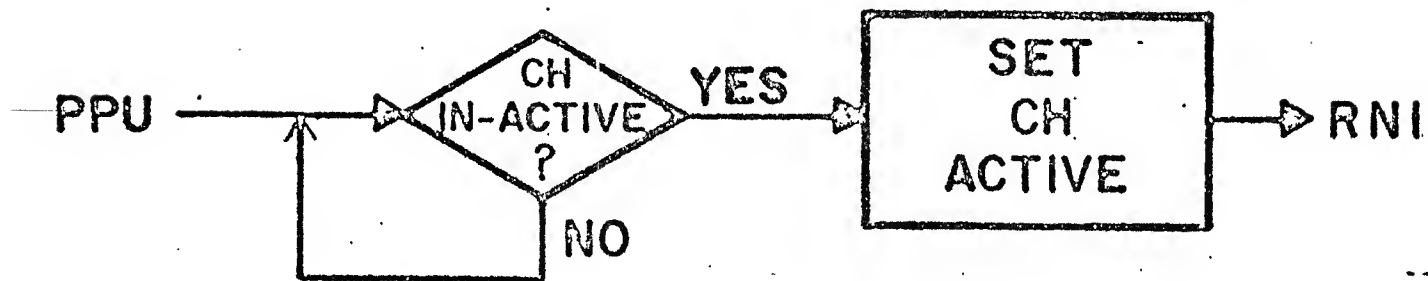
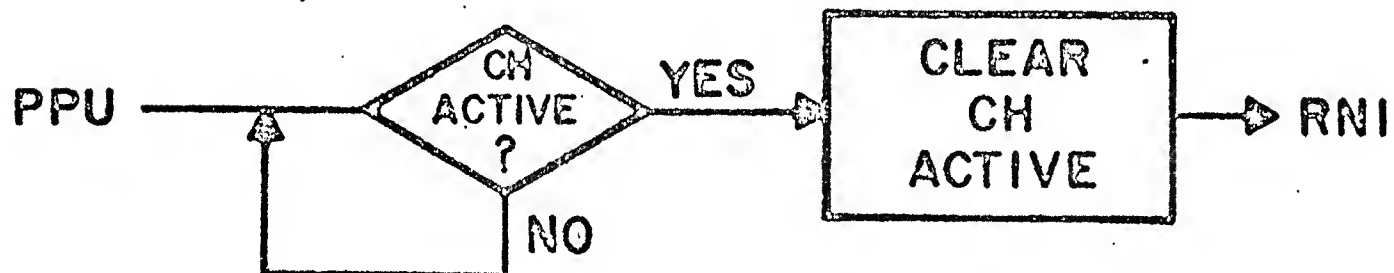


Figure 5-A-23. PPS Instruction 77

# ACN 74 ACTIVATE CH-d CHANNEL ACTIVITY

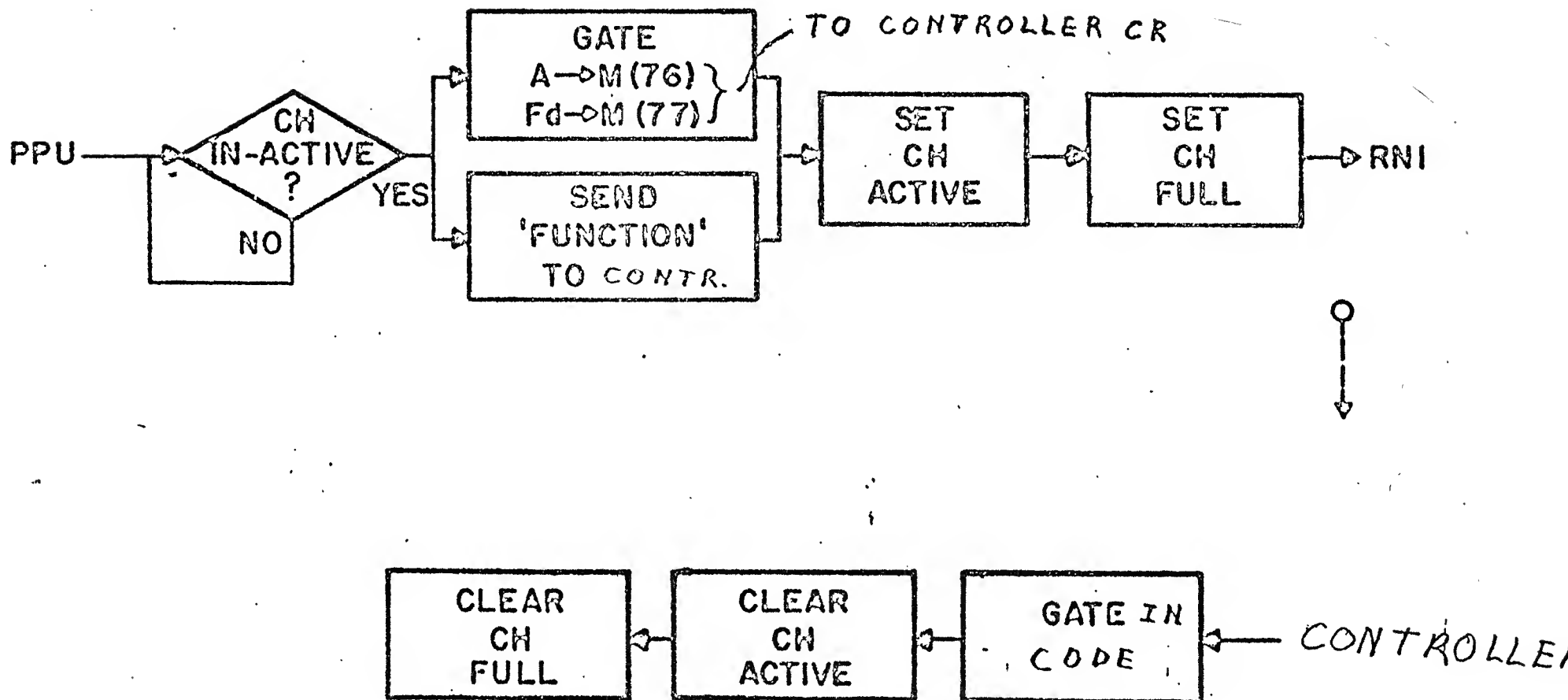


# DCN 75 DISCONNECT CH-d CHANNEL ACTIVITY

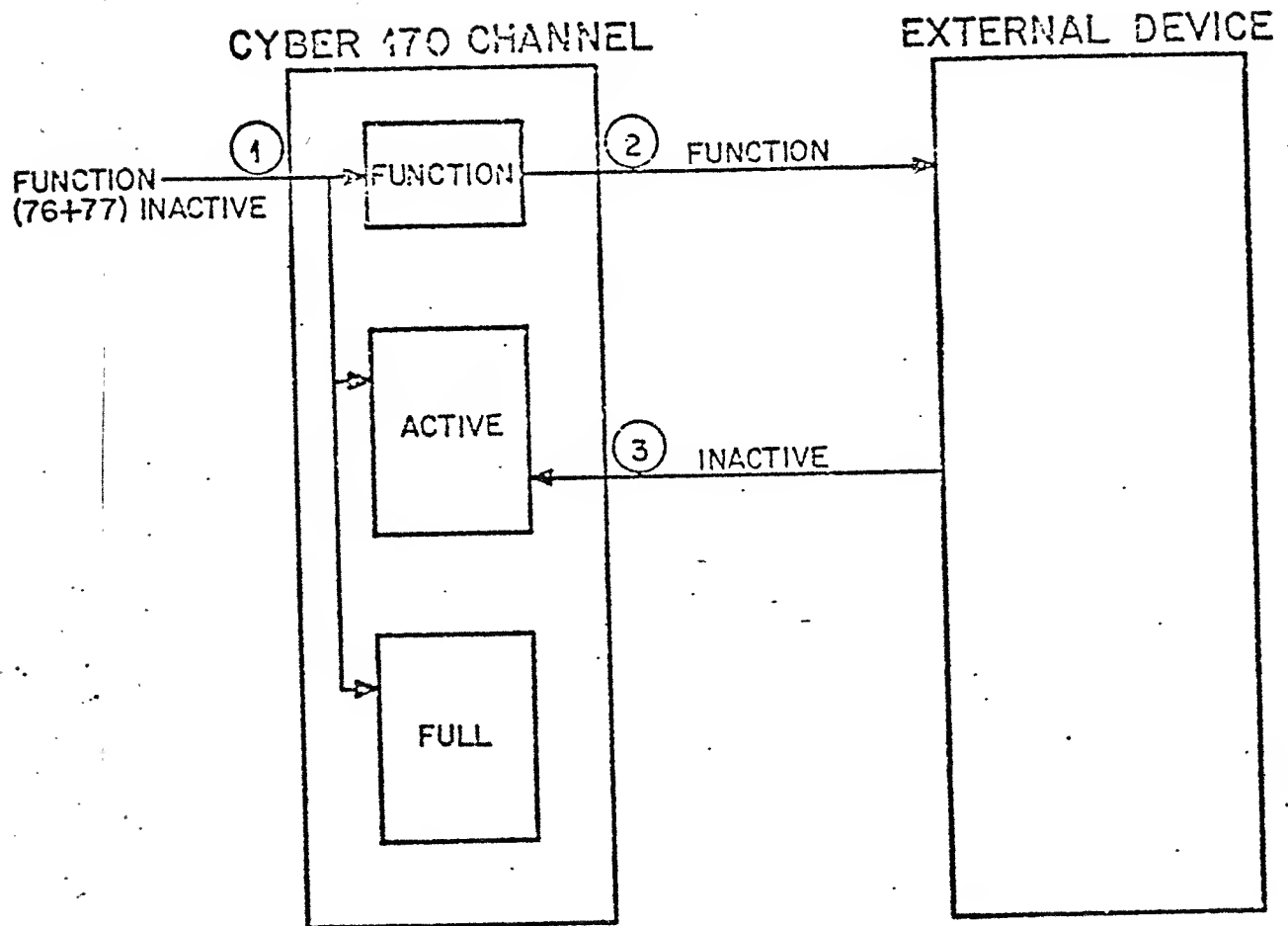


FAN 76 FUNCTION (A) ON CH-d  
FNC 77 FUNCTION m ON CH-d

CHANNEL ACTIVITY

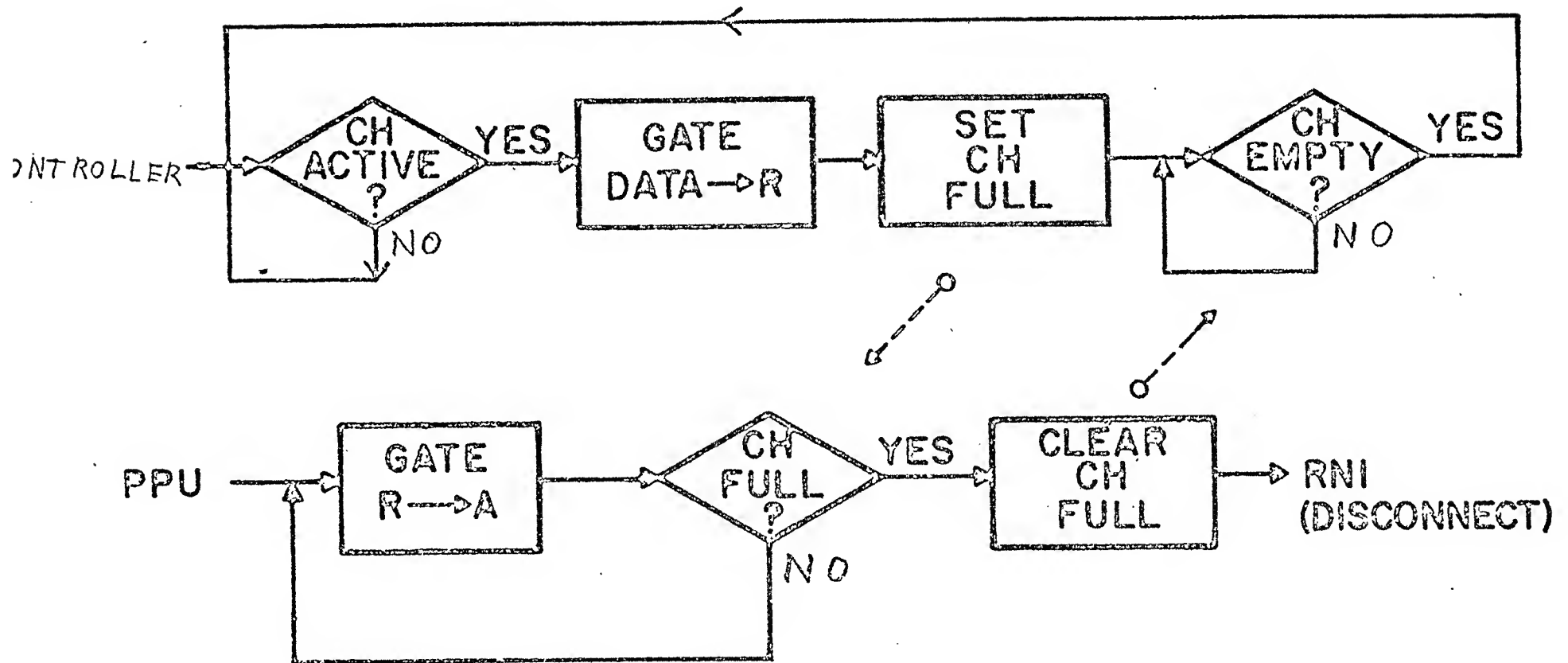


# EXTERNAL FUNCTION

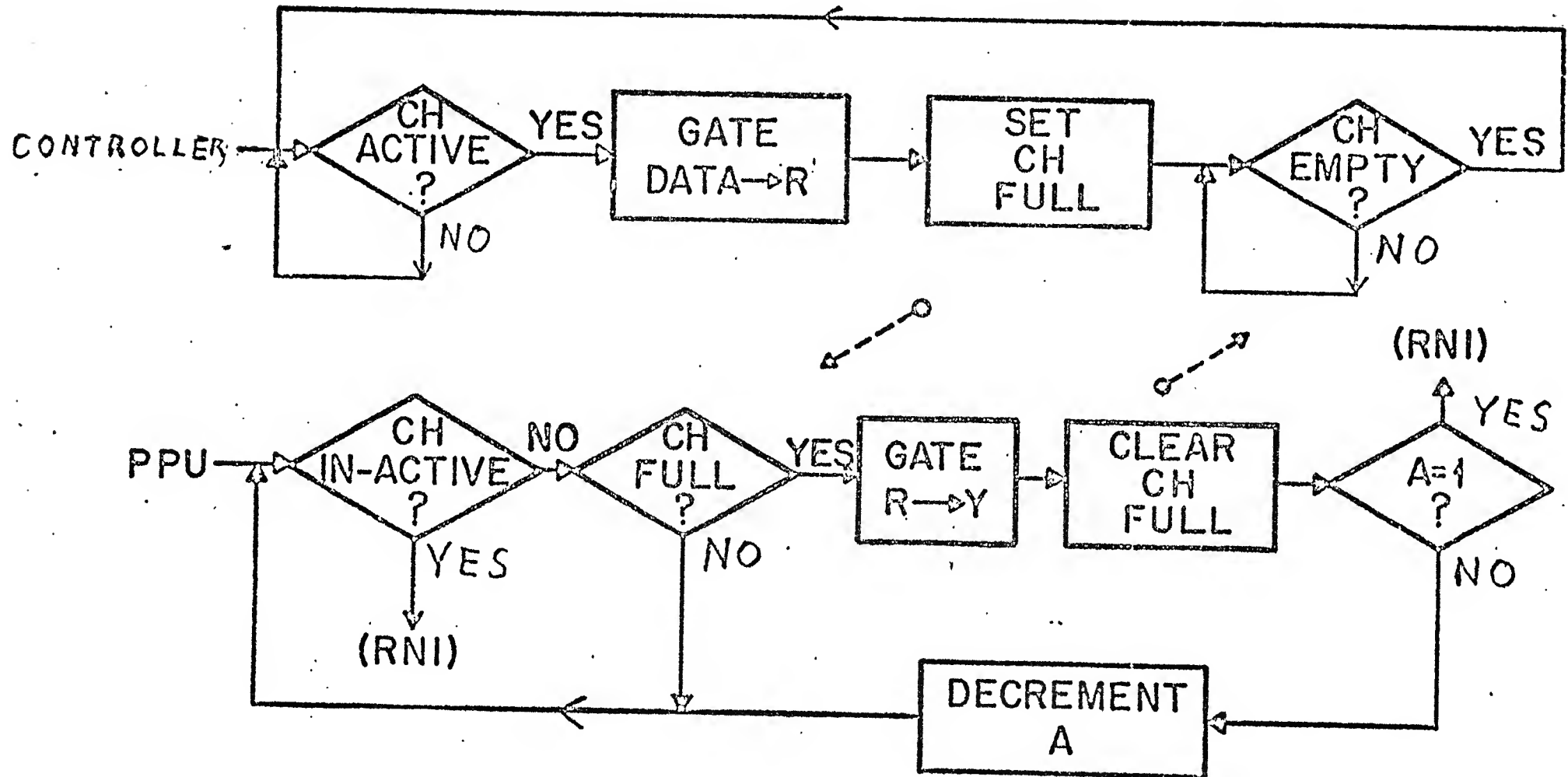


- ① Function from PP - result of 76d or 77d instruction.
- ② Function to external device - only if channel is inactive.
- ③ Inactive - response to function signal makes channel inactive.

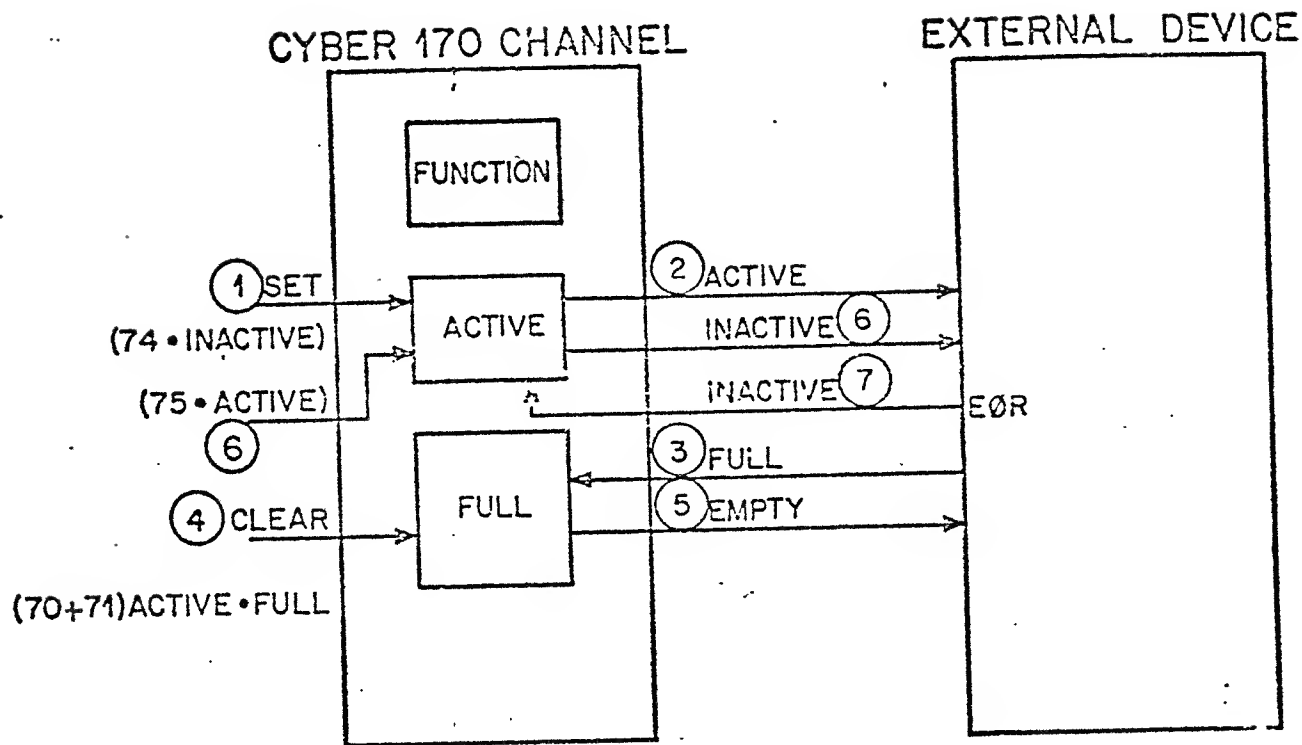
# IAN 70 INPUT TO A FROM CH-d CHANNEL ACTIVITY



# IAM 71 INPUT (A) WORDS TO $m$ FROM CH-d CHANNEL ACTIVITY

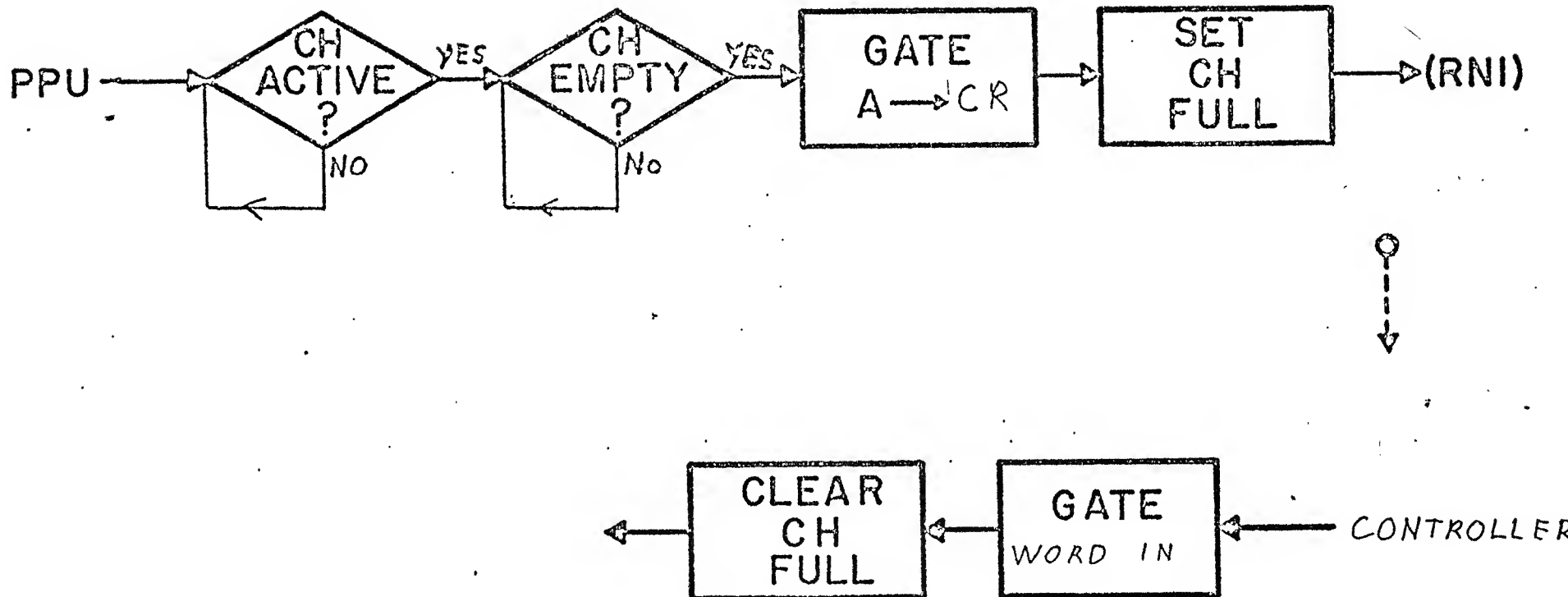


# EXTERNAL READ

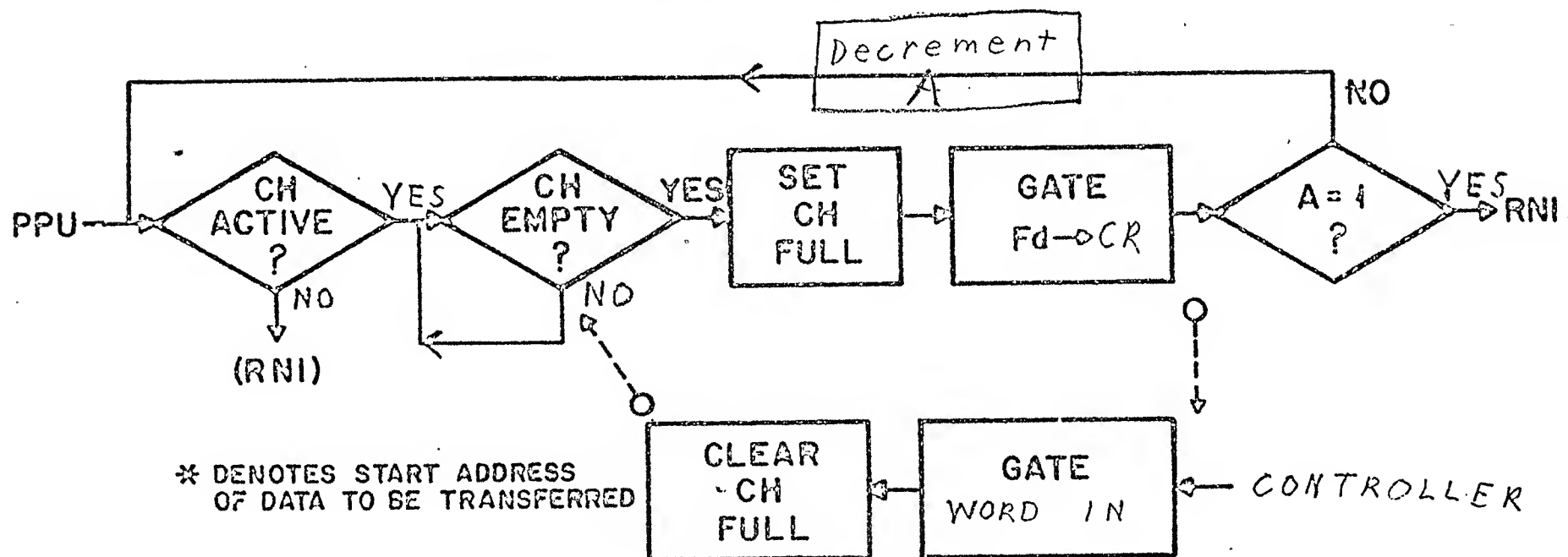


- ① Make channel active before requesting data.
- ② External device has been functioned previously to do a read. The active signal triggers the start of the operation.
- ③ With every data word inputted "full" goes set.
- ④ Clear the full F/F when the word has been put into "A" or PPU memory.
- ⑤ Empty sent to external device asking for another word. Goes back to ③ until:
- ⑥ (75 and active) clears active F/F when word count = 0.
- ⑦ Device has been functioned to read to end of record and "EOR" has been reached; clear active F/F.

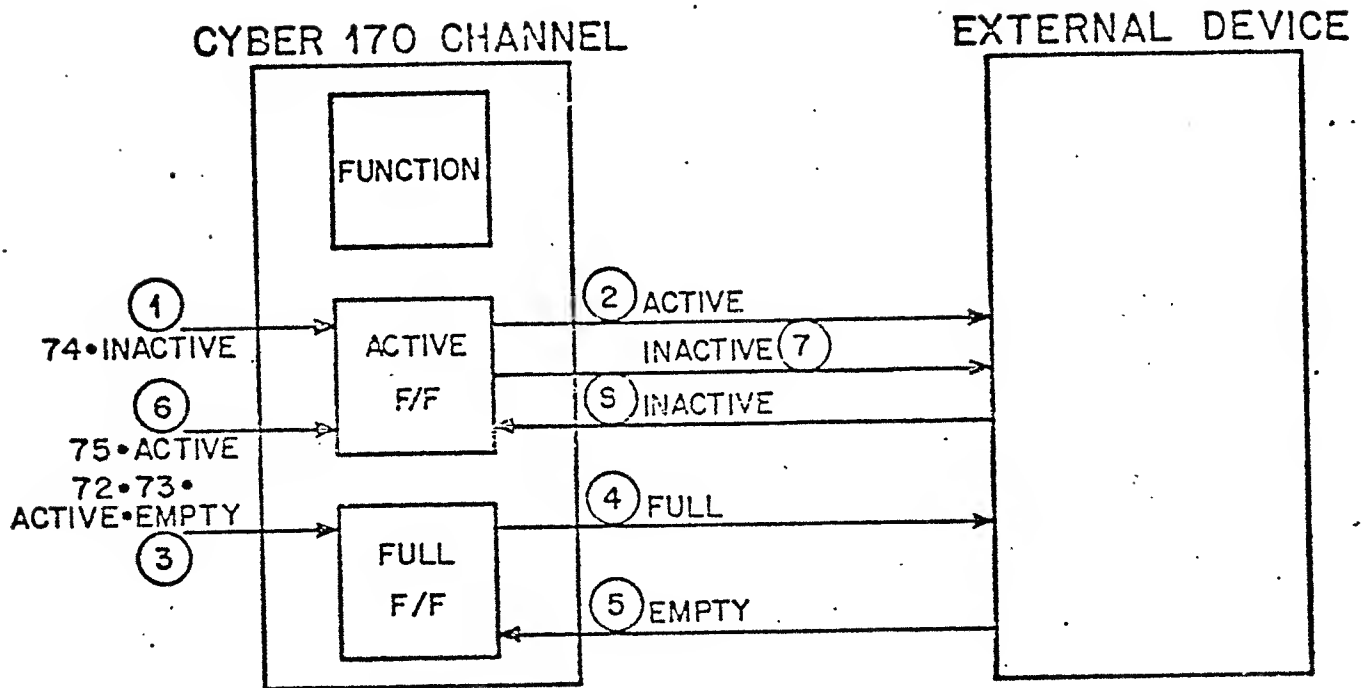
# OAN 72 OUTPUT FROM A ON CH-d CHANNEL ACTIVITY



# OAM 73 OUTPUT (A) WORDS FROM $m^*$ ON CH-d CHANNEL ACTIVITY



# EXTERNAL OUTPUT



External device has been functioned to do a write.

- ① (74d and inactive) - set active F/F.
- ② Send active signal to external device to trigger write operation.
- ③ 72 or 73 and active and empty - send data and set full F/F.
- ④ Send full to external device along with data.
- ⑤ Empty clears full F/F indicating receipt of data by the external device. Goes back to ③ until:
- ⑥ Word count = 0, clear active F/F.
- ⑦ Send inactive to external device to terminate the write operation.
- or ⑧ External device sends inactive usually as a result of an error condition (lost data, etc.).

DETAILED PAK DIAGRAM (PPS 3.11)  
CHANNEL CONTROL TRANSLATIONS

DATA INPUT SEQUENCE

An external device sends data to the PP as follows:

1. The PP places a function word in the channel register and sets the full and channel active flags. Coincidentally, it sends the word and a function signal to all synchronizers. The function signal tells all synchronizers to sample the word, and identifies the word as a function code rather than a data word. The code selects a synchronizer and a mode of operation. Nonselected synchronizers clear, leaving only the selected one turned on.
2. The synchronizer sends an inactive signal to the PP indicating acceptance of the function code. The signal drops the channel active flag, which drops the full flag and clears the channel register.
3. The PP sets the channel active flag and sends an active signal to the synchronizer. This signals the device to start sending data.
4. The device reads a word and then sends the word to the channel register with a full signal, which sets the channel full flag.
5. The PP stores the word, drops the full flag, and returns an empty signal indicating acceptance of the word. The device clears its data register and prepares to send the next word.
6. Steps 4 and 5 repeat for each word transferred.
7. At the end of transfer, the synchronizer clears its active condition flag and sends an inactive signal to the PP to indicate end of data. This signal clears the channel active flag to disconnect the synchronizer and the PP from the channel.
8. Alternatively, the PP may choose to disconnect from the channel before the device has sent all its data. The PP does this by dropping the active flag and sending an inactive signal to the synchronizer, which immediately clears its active condition flag and sends no more data. However, the device may continue to the end of its record or cycle; for example, a magnetic tape unit would continue to end of record and stop in the record gap.

DATA OUTPUT SEQUENCE

The PP sends data to an external device as follows:

1. The PP places a function word in the channel register, and sets the full and channel active flags. Coincidentally, it sends the word and a function signal to all devices. The function signal tells all synchronizers to sample the word, and identifies the word as a function code rather than a data word. The code selects a synchronizer and a mode of operation. Nonselected synchronizers clear, leaving only the selected one turned on.
2. The synchronizer sends an inactive signal to the PP indicating acceptance of the function code. The signal drops the channel active flag, which drops the full flag and clears the channel register.
3. The PP sets the channel active flag and sends an active signal to the synchronizer. This signals the device that data flow is starting.
4. The PP places a data word in the channel register and sets the full flag. Coincidentally, it sends the word and a full signal to the synchronizer.
5. The synchronizer accepts the word and sends an empty signal to the PP. This clears the channel register and drops the full flag.
6. Steps 4 and 5 repeat for each PP word.
7. After the last word is transferred and acknowledged by the synchronizer empty signal, the PP drops the channel active flag and sends an inactive signal to the synchronizer to turn it off.

The preceding material describes the sequence of events occurring on an I/O transfer.

The text for PPS detailed pak diagrams 3.11 and 3.12 contains a description of how channel control produces the control signals and flags necessary for I/O transfers.

PPS 3.11 shows those paks that translate instruction codes and channel status to enable the progress of I/O instructions. When I/O instructions arise in either an internal or external PP, the full/active control XLTR in the BW pak generates a selected GO signal to channel

control, thereby initiating a sequence of events that includes the alteration and transmission of channel status, as well as control of channel data in/out registers (PPS 3.6, 3.10). This sequencing is handled by hardware as shown in PPS 3.12.

An example of a GO signal is the output GO signal (BWOP). BWOP causes channel control to set the full flag, set data in the channel data in/out register, transmit full, and transmit data. Like the other GO signals, it is generated by either an internal instruction (732 + 720 . NF9) or an external instruction with priority (EXT WR. FLG).

The progress or delay of channel instructions depends on the channel's status. In the AV and AW paks, the status of the channel in operation is selected from the status of the other 13 channels by Q bits 0-3. AV uses QPRI bits 00-03 to select parity and status from one particular channel. Since QPRI is actually the channel designator for internal channel references and for references from the other PPS, the active and full signals are the status of any channel when it is performing I/O. As a result, these signals are used in the full/active control translator as well as in other PPS translators for status checks during I/O instructions 70-77.

Conditional jump instructions 64-67 require the active and full status of a specified channel regardless of which PPS is currently doing I/O; that is, one PPS may be interrogating the status of channel X by means of these instructions, while the other simultaneously has priority to do an I/O instruction on channel Y. The active and full signals are the status of channel Y in this case, and therefore could not be used for the conditional jump. Instead, internally generated jumps use AWFULL and AWACT1 (status selected by internal Q bits) while AWEFUL and AWEACT (selected by external Q bits in AW) are used in the other PPS as its external status signals.

#### PPS-0/PPS-1 DIFFERENCES

The PPS-1 real time clock (RTC-1) cannot be read. The logic level 1 applied to the AW pak forces channel 14<sub>8</sub> of PPS-1 permanently inactive and empty. RTC-0 differs in that it can be read on PPS-0 channel 14.

A large portion of PPS 3.11 shows hardware used to generate or receive requests, and give or reject priority for inter-PPS I/O instructions.

#### INTER-PPS I/O

This description uses PPS-0 as the source of inter-PPS I/O requests. In fact, either PPS may request the use of the other's channels.

PPS-0 specifies that it is addressing PPS-1 by setting bit 4 of the address contained in the Q barrel, thus identifying channels 20-33<sub>8</sub> as PPS-1 channels. If Q904 is present on an I/O instruction (70-77), the signal BXEXT is generated (BX module - PPS 3.11). This allows PPS-0 to send a request signal to PPS-1 for the use of its channels. The request is blocked on a skip or on the CWBLER (block external request) signal from the skip control circuit.

The external control translator (BS module - PPS 3.11) generates five signals of this type, asking a channel of PPS-1 to activate (BSCONN), to disconnect (BSDIEX), to write (BSWREX), to send a function (BSFCEX), or to read (BSRDEX). These signals arise on the appropriate instructions as shown in the translations of the BS module.

Figure 4-2-3 shows the sequence of events as they would occur on a 72 or 73 (write) instruction. BSWREX is the write request signal that goes through the five stages of the inter-PP control output register and a KT pak transmitter. It is transmitted to PPS-1 in the form of the EIWRIT signal.

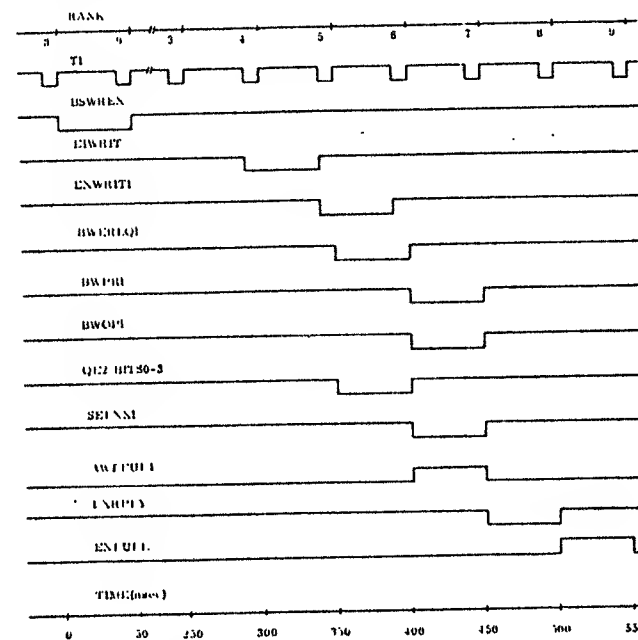


Figure 4-2-3. Inter-PPS Control Signals

When received at PP-1, E1WRIT is called E2WRIT. This is shifted out of the Inter-PP control input register as EXWRIT1, the suffix 1 indicating a PPS-1 signal.

If the request constituted by EXWRIT1 is accepted by PPS-1, the BWPRI1, BWREQ1 and BWOP1 signals are produced in the BW module. BWPRI1 and BWREQ1 indicate that the external request is accepted and BWOP1 initiates the output sequence. BWREQ1 gates in the channel number from PPS-0 in the QPRI1 selector.

BWREQ1 is received at PPS-0 in the form of E2RPLY, along with the E2ACTV and E2FULL status indicator signals. The inter-PP control input register of PPS-0 generates EXRPLY, EXFULL and EXAC from these signals. They are then used throughout PPS-0 instead of the full and active flags that would be used on internal I/O instructions.

Note that external requests receive priority when there are no internal requests ~~BLINREQ~~ or if an even major cycle is in process. Thus external requests are allowed every second major cycle even if continuous internal requests are in progress.

All external requests are processed in a similar manner, with a request going out and reply and status bits returning for use at the next slot time.

The data paths and control signals necessary in inter-PPS channel data transfers are further illustrated in figure 4-2-4. In this diagram, the PPS on the left is the requesting PPS and, therefore, all hardware pertaining to the origin of the request and reaction to the reply is shown. The depiction of the requested PPS (on the right) shows circuitry used in determining priority, selecting channel data; and sending data, controls and status back to the requesting PPS.

Note that the channel numbers, M barrel input and external request are all originated at the same slot time. In the next slot time (500 ns later), the reply, status and data are returned from the requested PPS.

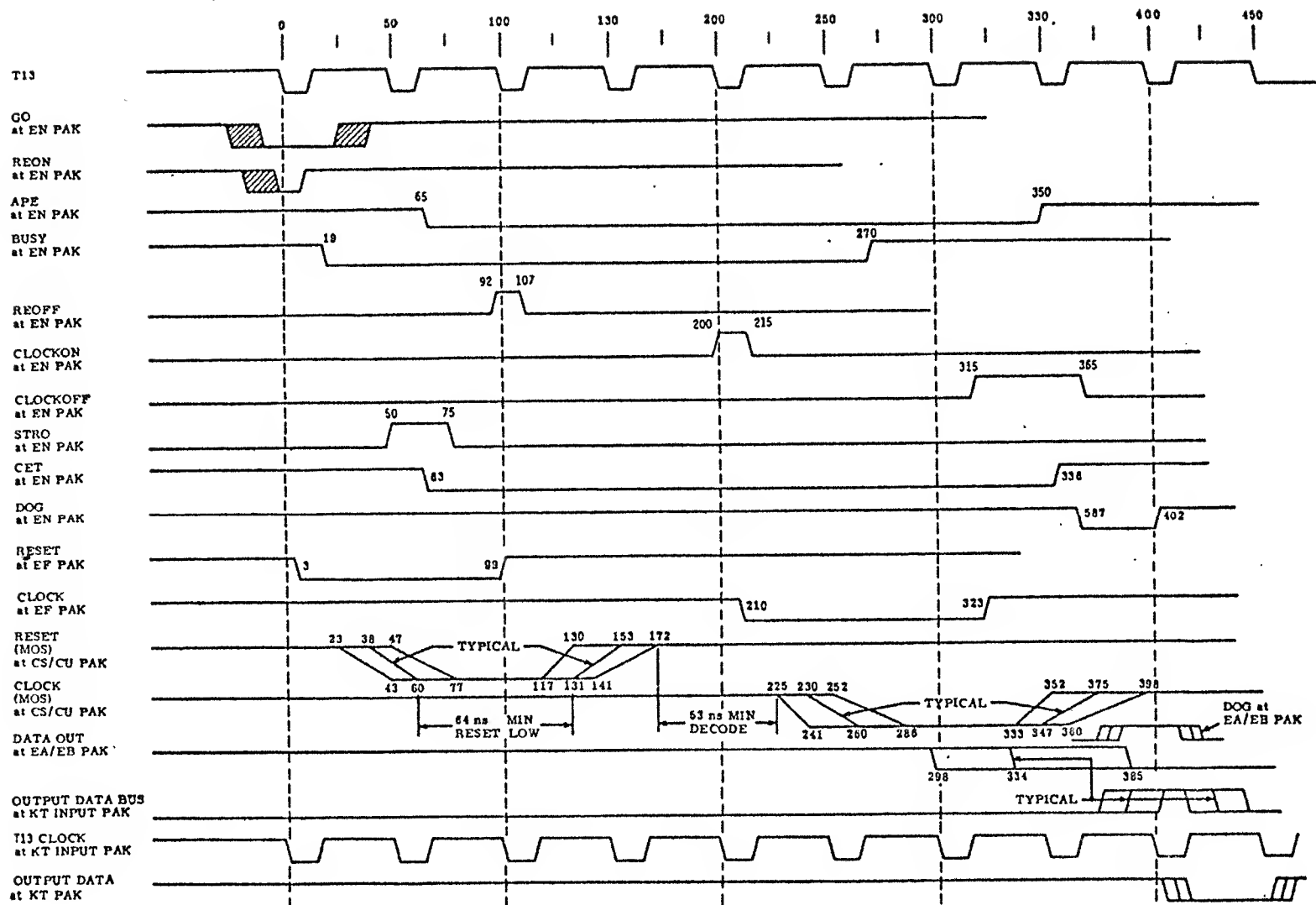


Figure 4-2-4. Inter-PPS Channel Data Transfers

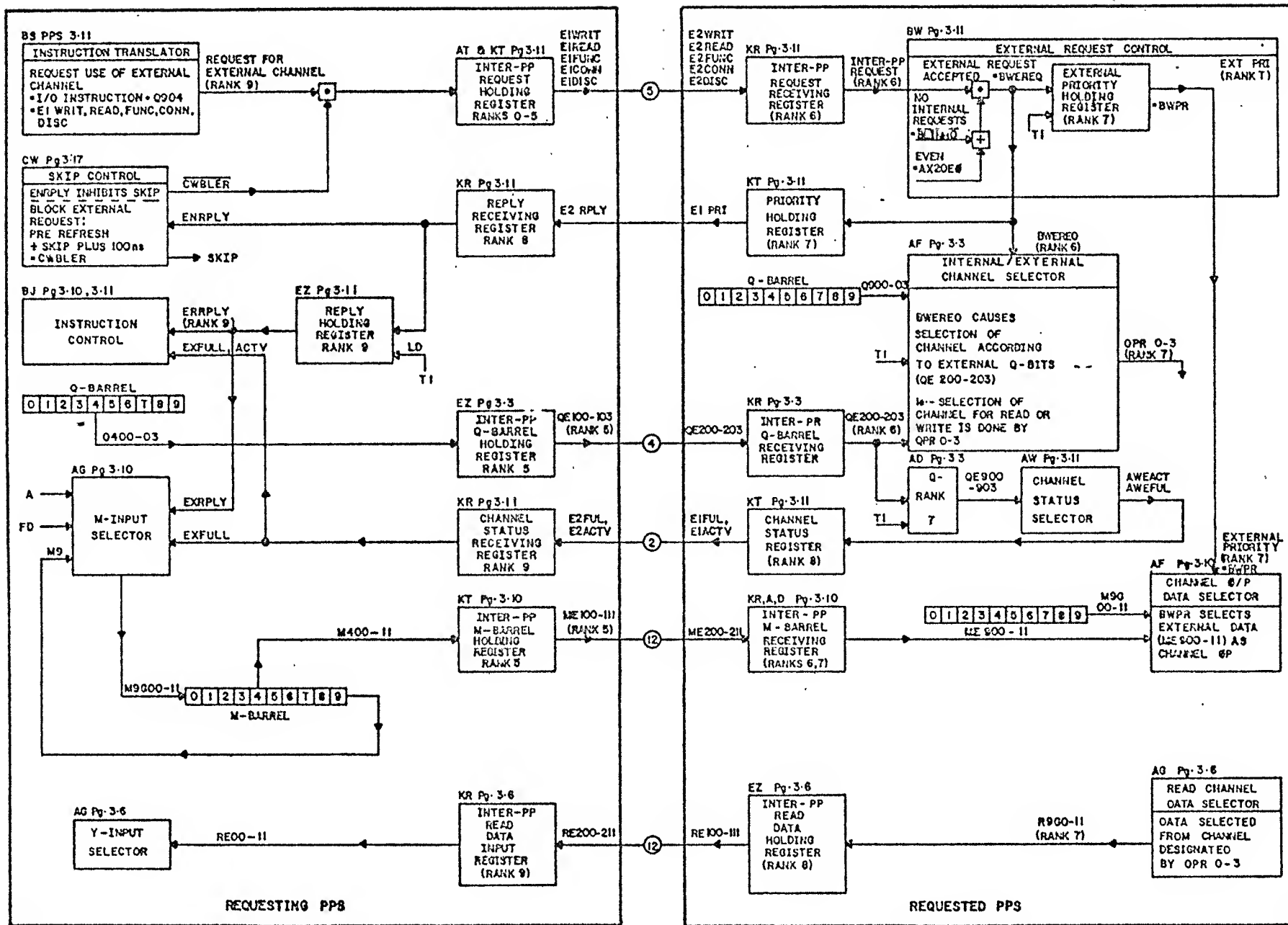


Figure 4-5-8. CSU Bank Timing.

# DETAILED PAK DIAGRAM (PPS 3.15) STATUS AND CONTROL REGISTER

The SCR is an array of bits used to indicate various status conditions and to set the mode of certain controllable operations. Many of these bits are exhibited in light emitting diode display modules (BZ paks - PPS 3.15). Specific bits assignments are shown in table 4-2-1.

Any bits that are forced set by error conditions are contained in the AS module. These error bits, as well as the other bits in the AS modules, are tested, set or cleared by means of output instructions 72 or 73. The data word (referred to for the purposes of the SCR as a descriptor word) arriving at the SCR with these instructions specifies the address of a bit and the operation to be performed on this designated bit. The SCR is permanently wired on channel 16<sub>g</sub>. It is accessed and controlled by the PPS through this route.

An explanation of the descriptor word bit assignments and the various functions available are shown in tables 4-2-2 and 4-2-3, respectively.

## CONTROL SIGNALS

The origins of SCR control signals and their functions are as follows:

CHER00-13	A parity error on a particular channel is indicated. The corresponding SCR bit is set immediately on activation of one of these signals.
PMER00-13	A parity error from a particular PPM is indicated. The SCR bit sets upon triggering by this signal.
AGYP0-23	One of these bits is activated on command from channel 16 <sub>g</sub> to select one group of 8 bits in the AS module.
MST0-2	This is part of the descriptor word bit address. It specifies a bit within a group for manipulation and/or testing.
ZSET	This signal enables the setting of a bit specified by AGYP and MST. It is generated on the set instructions (4XXX and 5XXX).

LDST	Sets the specified bit if ZSET is active. If <u>ZSET</u> , the bit is cleared. Whichever action is specified, it occurs on the trailing edge of LDST. LDST is only produced on the 2-5 function codes which either set or clear a specified bit.
AZCA	The activation of this level causes immediate clearing of all SCR bits in the AS module.
AZDSCL	A signal produced by SCR control (PPS 3.16) to clear certain SCR bits on DSRT.
SST bits	Any control signal, SST xx forces STC xx to a 1. The PPS generates SST 00 on a CM read parity error to set STC 00.

## LATCHED BITS

Within the AD modules are those status bits that may only be set upon gating by other SCR bits or on the PLATCH signal. All transitions occur on the trailing edge of the gating signal.

## THE SCR IN PPS-1

Physically, the status and control register in PPS-1 (called SCR-1) is almost identical to SCR-0. SCR-1 fills the same pak locations as SCR-0. The difference between the two is that SCR-0 makes use of many more bits than SCR-1 through interconnections with other units. Those used on SCR-1 (channel 36<sub>g</sub>) are listed in table 4-2-1.

SCR bits 036, 037 are used in PPS-0 to indicate power failures or imminent power failures, respectively. In PPS-1, they are not connected to any warning signal and therefore serve no function. To prevent these bits from being forced set and causing a PPS-1 error condition (AZSYER), SST 036, 037 are forced to zero.

In either PPS chassis, SSTS10 indicates an SCR error condition in the other PPS. SSTS10 is disabled in PPS-1 to prevent SCR-1 bit 10 from setting on an SCR-0 error condition.

Only those status and control bits pertaining to PPS-1 register in SCR-1. As a result, the only interface between SCR-1 and other chassis is the PPS-1 breakpoint match signal (BKPP0). With the exception of SSTS10, all interchassis signals to or from SCR-1 are made ineffective by omitting cable connections, or by disregarding them at the CPU and CMC.

TABLE 4-2-2. DESCRIPTOR WORD

FUNCTION CODE		NOT USED	SCR BINARY ADDRESS								
11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-7 are the BINARY SCR ADDRESS. They are decoded to select a particular SCR bit or word.

Bits 9-11 are the instruction code in binary form. The function designated by each code is shown in table 4-2-3.

NOTE that the descriptor word is MJ00-11. These bits are taken directly from the M input selector (PPS 3.10).

TABLE 4-2-3. FUNCTION CODES

CODE	FUNCTION	REMARKS
0	READ	Places the designated 12-bit word on Channel 16 <sub>g</sub> .*
1	TEST	Places the designated bit on Channel 16 <sub>g</sub> .
2	CLEAR	Resets the designated bit.
3	TEST/CLEAR	Places the designated bit on Channel 16 <sub>g</sub> , then resets bit.
4	SET	Sets the designated bit.
5	TEST/SET	Places the designated bit on Channel 16 <sub>g</sub> , then sets the bit.
6	CLEAR ALL	Clears all bits.
7	TEST ERROR	Places a 1 in lowest order bit of Channel 16 <sub>g</sub> if any error bits (bits 0-39 <sub>10</sub> ) are set.

\*On the read function, descriptor word bits 0-4 designate an SCR word location.

On all other functions, descriptor word bits 0-7 designate a specific SCR bit.

DETAILED PAK DIAGRAM (PPS 3.16)  
STATUS SELECTOR - STATUS AND CONTROL REGISTER

The three stages of status input selectors are shown in PPS 3.16.

The first two stages select an STC word for input to channel 16<sub>g</sub> according to MST0-2 and AYM13. AYM13 is a fan-out of MST03.

Status input selector 3 normally outputs the AZTEST bit. On a 1, 3 or 5 function code (single bit polarity test functions), AZTEST is the same polarity as the bit being checked. On a 7 function code, AZTEST is a 1 if one or more of the error indicator bits within the SCR are set.

On a zero (0) function code, the signals AZOP and AZREAD are generated. On this combination status input selector 3 inputs the word selected at status input selector 2 to channel 16<sub>g</sub>.

DETAILED PAK DIAGRAM (PPS 3.17)  
CONTROLS FOR STATUS AND CONTROL REGISTER

The AZ module provides the control signals for the status and control register. (See PPS 3.15; the functions of these signals are described in the accompanying text.)

Control signals AZCLRA, AZLDST, AZSET1 and AZREAD are generated according to both MJ09-11, which is the function code portion of the SCR descriptor word and the AZOP signal.

AZTEST is activated on the test functions (codes 1, 3, 5, 7) which are also decoded from MJ09-11. On a 7 code, AZTEST activates to indicate the presence of an SCR error bit. On the other test functions (codes 1, 3, 5), AZTEST is set only if the particular bit under test is a one. This condition is indicated by the absence of either STT02 or STT03.

The channel 16 full status controls generate AZOP for 50 ns on an output instruction to channel 16. BWOP, if gated in by the select channel 16<sub>g</sub> signal (SEL16), sets the status full control flip-flop. This, in turn, sets status full register 1 which activates AZOP. The inverted output of status register 2 is a necessary condition for AZOP. Therefore, when status full register 2 sets 50 ns after status full register 1, AZOP is deactivated.

AZOP is a necessary condition for carrying out any SCR function code.

Setting the status full control flip-flop also enables translation of the MJ03-07 bits to generate AYGP group select bits. This also gates the current MJ00-02 bits to the SCR to be used as an individual bit address or as a portion of a word address.

AYM13, which is a fan-out of MST03, forms the most significant bit of the SCR word address.

PPS-1 sends a system error signal (E1PPER) to PPS-0 to activate SCR-0 bit 10<sub>10</sub>, thereby indicating an error in PPS-1. PPS-0 also sends such a signal to PPS-1 which is received as E2PPER. This signal is not used and, therefore, the SST010 signal is disabled in PPS-1.

On an input instruction to channel 16<sub>g</sub> the BWMT signal is generated by channel control and enabled by the SEL 16 signal. This resets the status full control flip-flop to allow another PP to access channel 16<sub>g</sub>, while at the same time inputting data from status input selector 3 (PPS 3.16). Thus a complete channel 16<sub>g</sub> access cycle must include an output instruction followed by an input instruction.

## DETAILED PAK DIAGRAM (PPS 3.18)

### PP CODE COUNTER

A PP code counter (AX module), which increments on every T1, and resets on its nine count, produces the signals AXCY0-3. Together, these signals are a binary representation of the number of the PP in the slot at any given time.

One function of the AXCY0-3 bits is to enable one of PMER00-09 on the receipt of a PPM parity error signal (BPFMST). The selected PMER signal then sets the SCR bit assigned to a parity error for the PPM involved in the error.

Note that in PPS-1 certain PP memory error bits are disabled by a logic zero level. This is done for all parity error signals corresponding to nonexistent PPMs. Thus PMER04-09 is inhibited in a 14-PP system, whereas only PMER07-09 is disabled in a 17-PP system. Naturally, all these bits are enabled in a 20-PP system.

The disabling of appropriate PPM parity error bits prevents the setting of SCR-1 error bits under false error conditions.

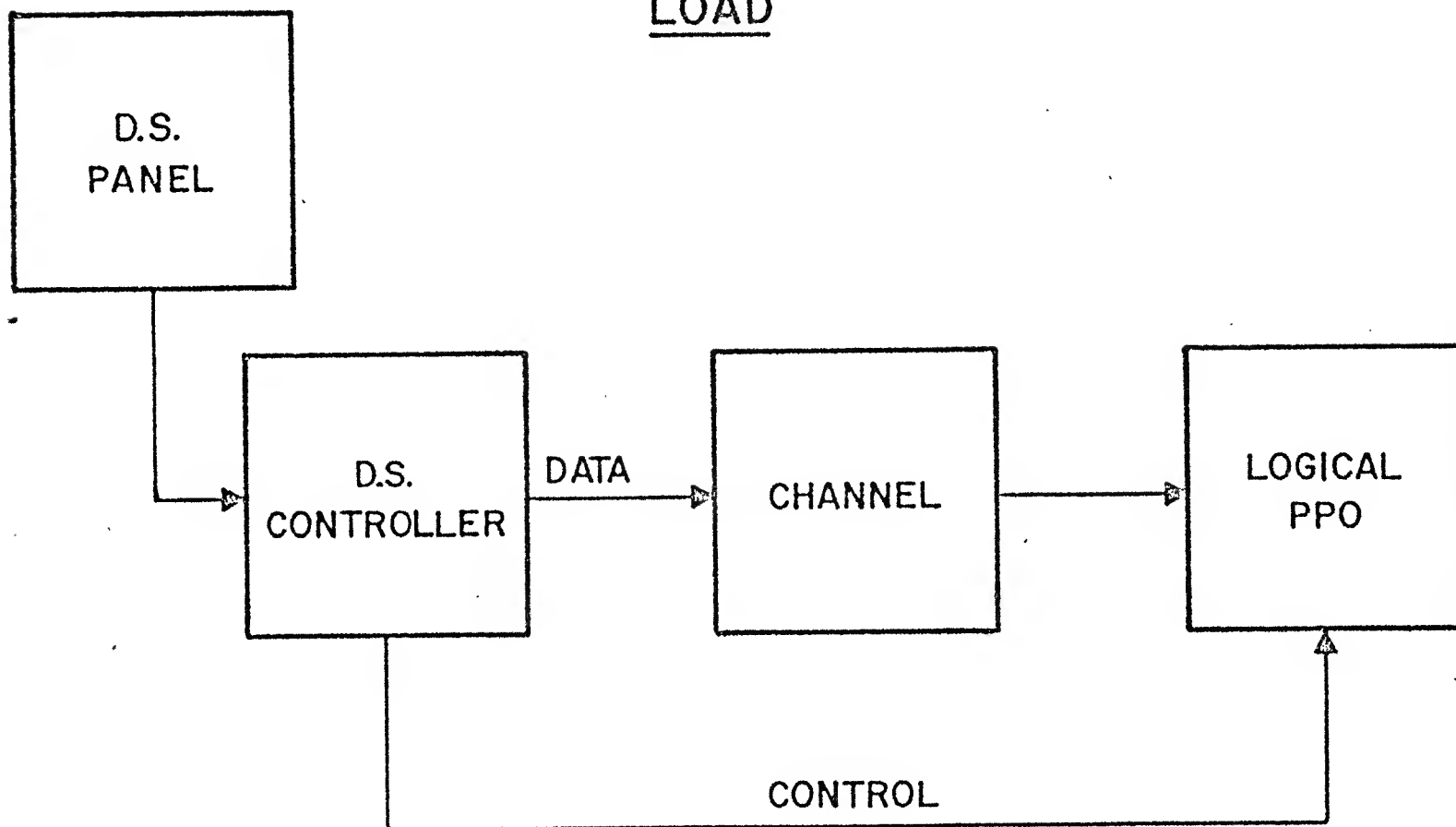
The major cycle signal (AXMAJC) occurs every 500 ns for a duration of 50 ns on each 9 count of the PP code counter. In alternating major cycles, the even/odd flip-flop produces the AXODD and EVEN signals. From the AXMAJC, AXODD and EVEN signals are generated AXMJE and AXMJC. All these signals are used for PPS timing purposes.

The early refresh flip-flop produces the REFRF7 signal during a refresh cycle (indicated by CL32) on a PP count of 9. This is the last minor cycle of a 500-ns refresh cycle during which PP-0 is in rank 8. REFRF7 inhibits all changes in PP-0's A and Q registers in rank 8. Any such changes would be undesirable as they would arise from K8 translations produced from the meaningless memory references performed during refresh.

Synchronization of the PP code generator is controlled by the P2SYNC and P2SYNT signals. P2SYNT, a pulse-widened fan-out of T20, fixes the phase relationship of the 100-ns clocks to the PP count so an even-numbered PP is always in the slot at T20 time. This provides a fixed timing relationship between the 100-ns and 1000-ns channel clocks. (See figure 4-2-9.)

The two PP counters of the PPSs are also synchronized to each other to have identical counts at any given time. This is accomplished by sending a 50-ns pulse from PPS-0 to PPS-1 on the 9 count of every even cycle. As shown in figure 4-2-9, this pulse is called AXMJE where it leaves the AX psk. After a 100-ns delay in transmission, it goes to the AX pak in PPS-1 to force a PP count of 2, thereby synchronizing it with the PPS-0 count which would be going to a 2 at the same time. Figure 4-2-9 shows the paks involved in a strictly functional representation. It also shows the inter-PPS real time clock synchronization. The PP counter in PPS-1 is a slave to the one in PPS-0. In model A, this is accomplished with identical hardware in PPS-0 and PPS-1 except that a ZERO level inhibits the SYNC signal in PPS-0. Model B simply replaces the SYNC signal with a ZERO. The wiring differences needed to establish the master-slave relationship are shown in PPS 3.15.

## DEAD START LOAD BLOCK DIAGRAM

LOAD

## DETAILED PAK DIAGRAM (PPS 3.19)

### DEAD START AND SKIP CONTROLS

#### SKIP CONTROLS

The mechanism for generating the skip signal as well as other related control signals is contained in the CW module.

The skip level prevents any changes in a PP's registers during slot time.

Two conditions generate a skip. The first is indicated in the CW module diagram as LXSPD (one times speed). When the PPs are to operate at half their maximum speed, STC084 and consequently 1XSPD is activated. Note that circuitry is provided in CW to control PP speed by either STC084 or EX2XSP. In PPS-0, STC084 is enabled for this function; whereas in PPS-1, EX2XSP (the name given to STC084 when received in PPS-1) performs speed control.

1XSPD forces a skip on every even cycle as long as neither of the two skip inhibiting conditions exist. One of these conditions is a reply for an external channel instruction. Such a reply, as indicated by RPLY 9.Q904PR, must inhibit skip to prevent loss of data from external channels.

The other skip inhibiting condition arises on both the 604 and 605 trips of the 60 instruction. Skip is inhibited on these trips in order to yield a 6-trip instruction for compatibility with the CYBER 70 series.

The other skip-forcing condition arises on a refresh cycle. Timing for such a cycle is illustrated in PPS 3.9. The PPMREF signal sets the skip holding I & II flip-flops on successive major cycles with the result that REFSKIP activates skip on the major cycle following CL32, as shown in the timing diagram.

The forcing of skip under refresh conditions prevents use of erroneous PPM data generated during the refresh cycle. At the same time, the correct data is read from PPM.

In the case where a PP is doing a PPM write during the refresh cycle, the write takes priority over the refresh. The refresh then occurs in the skip cycle, which is generated as described above.

The CWBLPC signal blocks the generation of a PPM parity error signal.

One condition forcing CWBLPC is the 5x7 holding FF. On a PPM parity error, the force K circuits (PPS 3.5) force K to 5x6 in order to hang up the offending PP. This in turn, sets the 5x7 holding FF. CWBLPC prevents the setting of the PPM parity error bit in the SCR if it is reset by software while the PP is still hung up.

Any data obtained from PPM during a refresh cycle will be erroneous. Since skip produces CWBLPC, and refresh happens during skip on a read memory cycle, any such erroneous data will not cause a parity error on a read cycle. However, if a write cycle is in progress when the refresh (CL32) signal arises, the refresh is delayed to occur after the skip cycle. Under these conditions, the ATRD19 signal is generated by the initial write signal time to block the parity check during delayed refresh.

CWBLER blocks any requests to the other PPS (external requests). It arises on the major cycle preceding a refresh skip so no responses will return from the other PPS during refresh skip time. EXRQB2 also generates CWBLER, thereby inhibiting any requests which would arrive during the active cycle of the other PPS. This ensures that priority will be received by external requests.

#### DEAD START CONTROLS - GENERAL DEAD START

The mechanism for producing dead start signals (BADSRT and DSRT) is shown in the BA pak (PPS 3.19). The real time counter (RTC CNTR), clocked by MJE.T1, counts to 7777<sub>8</sub> and then resets. For the duration of the 7777<sub>8</sub> count (1  $\mu$ sec), the decoder produces the BADSSN signal.

If one of the dead start push-buttons is pressed as indicated by DSPBTN, or if the dead start switch is on (DSWICH), DSRT FF1 sets enabling the BADSRT signal. BADSRT activates for the duration of the next BADSSN producing a 1  $\mu$ sec dead start period.

After this dead start period DSRT FF2 resets, thus disabling BADSRT as long as the dead start push-button is pressed. However, if the dead start switch initiates dead start, this disabling action is bypassed and a continuous dead start results.

On a continuous dead start, the BA pak produces DSRT every 10000<sub>8</sub> (4096  $\cdot 10$ ) microseconds until the dead start switch is turned off.

Timing for a single-shot dead start is shown in figure 4-2-9.

Controls from the dead start panel connect in parallel to the BA paks in both PPSs. Since the RTCs in these paks are synchronized (see text below), as are their major cycle clocks (see text for PPS 3.15), the dead start signal (DSRT) arises simultaneously in both PPS chassis. In PPS-1, the production of DSRT is the only function of the RTC. This RTC is not read as a channel input in the way that the one in PPS-0 is read on channel 14<sub>g</sub>.

As noted above, the RTCs of two PPSs are synchronized. This is carried out in a master-slave relationship between PPS-0 and PPS-1, respectively. As illustrated in figure 4-2-8, the PPS-0 RTC sends the BADSSN signal to PPS-1 during its maximum count of 7777<sub>g</sub>. The RTC in PPS-1 receives this signal 100 ns later and is forced to a count of zero on the next AXMJE signal as will be the PPS-0 RTC at the same time. The synchronized count is further facilitated by the synchronization of AXMJE signals between PPS chassis.

Figure 4-2-8 shows the two PPSs in a strictly functional representation. In fact, both PPSs contain identical hardware except that the real time clock synchronization pulse (EXRTCS) is inhibited in PPS-0 by a logic zero level, thereby making PPS-0 the master and PPS-1 the slave. This is illustrated in PPS 3.19.

The BADSRT and DSRT signals are fanned out to cause the following actions during dead start:

- 0 Sets all channels to the active and empty conditions by means of the SACT00-13 and ARMT00-13 signals.
- 1 Generates the ADSCL (PPS 3.17) to clear certain SCR bits.
- 2 Resets status flip-flops in the BG module and inhibits any CMC requests (PPS 3.14).
- 3 Resets the memory refresh address counter and sets the memory sequence decade counter to the PPM number designated by the PPM select switches (PPS 3.9).
- 4 Resets read pyramid status flip-flops (BK pak, PPS 3.13).
- 5 Sets peripheral equipment, CMC and CPU to initial conditions by means of master clear signals.
- 6 Forces P → G by resetting the G input control holding register (PPS 3.8).

BADSRT is also fanned out in the form of CX17DS and CX50DS to carry out the following functions:

- 7 Sets K and inhibits exits (PPS 3.5). The value to which K is forced depends on the dead start mode select switch. The three possible modes, the conditions signalling the mode at the force K circuits (PPS 3.5), and the value to which the force K circuits set K are as follows:

Load occurs on DSRT . DSRT1 . DSRT2 forcing K = 710  
 Sweep occurs on DSRT . DSRT1 . ~~DSRT2~~ forcing K = 505  
 Dump occurs on DSRT . DSRT1 . DSRT2 forcing K = 730  
 In the load mode, DSRT produces the BBLOAD signal.

- 8 Forces P → Y (PPS 3.6).
- 9 Forces the A register to 10000<sub>g</sub> (PPS 3.1).
- 10 Loads the PP numbers into the Q barrel to assign each PP to the corresponding channel number (PPS 3.2). Note that DSRT is synchronized with the PP numbers because it is initialized on the minor cycle following AXMAJC. AXMAJC occurs only when PP9 is in the slot. (See BA and AX modules - PPS 3.19 and 3.18).
- 11 Resets rank 9 of the P register (PPS 3.4).

#### LOADING FROM THE DEAD START PANEL

The sequence of events loading locations 0-21<sub>g</sub> in PPM0 is initiated in the CY pak (PPS 3.19) by DSRT. Timing for the control signals involved in this process is shown in figure 4-2-10. The CY pak produces these signals.

All channels are initially set to active and empty on DSRT. However, the status of channel 0 is forced to full by means of the CYNF00 signal which arises on DSRT. At the same time the CYDS1 signal sets. This gates the current dead start panel word into the Y register (PPS 3.6) rather than the normal channel 0 bits (R00 bits 0-11). BBLOAD activates the CYDS00 signal which forces the DSRT panel data word to zero. It also sets the CYSL counter to zero.

The termination of BBLOAD is coincidental with that of DSRT. At the instant of termination, the 71 instruction proceeds freely. The sequence is as follows:

- 710 - P → G. Since P was set to zero on DSRT, PP-0 addresses location zero which was set to zero during DSRT.
- 711 - FD-1 → P meaning that 7777 → P → G.
- 712 - R → Y. Since CYDS00 and CYDS1 are active on the first 712 trip, R = 0 and therefore zero → Y. The combination of K = 712, CYNF00 and CYNA00 causes a PPM write. The address for this write is location zero because a 712 instruction forces a P increment before P<sub>0</sub> → G.

On this and successive 712 instructions the BWMT signal is produced on P1-0 slot time because channel 0 is forced full. Through a chain of flip-flops (BWMT = CYDFP → CYDS00 → COUNT), the count clock pulse is produced

causing the CYSL counter to increment. A decoding and conversion network sends the new count to the dead start panel which responds by incrementing the dead start panel word address.

The CYSL count = 16 condition (CNT16) activates the CYDS00 signal for the second time and consequently PPM control loads a zero from the DSRT panel into location 21<sub>8</sub>.

This forces a K increment to 713 with the result that the 71 instruction completes and exits in the normal fashion. Channel 0 remains inactive and empty when enable DS, CACT00 disconnect and CYDS1 reset on disconnect.

The result of the 713 and 714 instructions is that the 71 instruction exits to the dead start instruction in location one.

The dead start control translator is altered in PPS-1 to prevent the latching of the enable DS signal. This enables channel 00 of PPS-1 to assume the status (active and empty) of a normal I/O channel at dead start. If enable DS was enabled in PPS-1, PP-0 would step through the 71 instruction in an attempt to read a nonexistent dead start panel.

A further change made in PPS-1 is the removal of two of the three CZ paks used for interface with the dead start panel because they have no function in PPS-1. The third CZ pak, in location E06, is used as an external full receiver (PPS 3.10).

#### PP SELECTION

The CX module (PPS 3.19) contains circuitry to select a particular PP for various purposes. This circuitry provides three ways of selecting a PP:

1. Breakpoint mode - enabled by STC083. Under normal circumstances, with STC083 set, the BGA0 signal (constituting a request to CMC) produces a 50-ns PLATCH signal which locks the contents of the requesting PP's P register, as well as its PP number, into assigned bits in the SCR. On a breakpoint match caused by

a CM access by a PP, CMC sets STC076. STC076 inhibits the PLATCH signal from occurring on successive CMC requests. The P register contents and PP number of the offending PP are latched into the SCR until software resets STC076. Timing for this mode is shown in figure 4-2-11.

2. Manual display mode - enabled by STC083 . STC124. In this mode, the PP is selected according to manual PP select switches (switchpak, location J40). The counter and latch circuit within the CX module produces a 17-ns PLATCH signal during the slot time of the selected PP. This latches the PP number and P register into SCR bits 60-75<sub>10</sub>.
3. Automatic mode - enabled by STC083 . STC124. Same as manual mode except that PP selection is according to STC 120-123 rather than the PP select switches.

Methods 2 and 3 of selecting a PP are used to force the exit of a single PP or to dead start a single PP.

STC bit 125 enables the forced exit of a single PP. The SELPP signal is produced for the full 50-ns slot time of the selected PP. During this period, the CXEXIT signal is active. It forces an exit in the exit control translator (PPS 3.5). A latch circuit prevents the regeneration of CXEXIT until software clears, and then sets STC125.

STC 126 enables the dead start of a single PP. The CX17DS and CX50DS signals are activated for 50 ns when the selected PP is in the pre-slot and slot, respectively. Note that actions 0 to 6 listed as occurring on a general dead start (see above) do not occur on a single PP dead start.

Timing for a single PP dead start is shown in figure 4-2-12.

Note that CXEXIT is a one-shot signal, there being a latch mechanism preventing another signal until the controlling SCR bit is reset and set again. CX17DS and CX50DS may occur repeatedly.

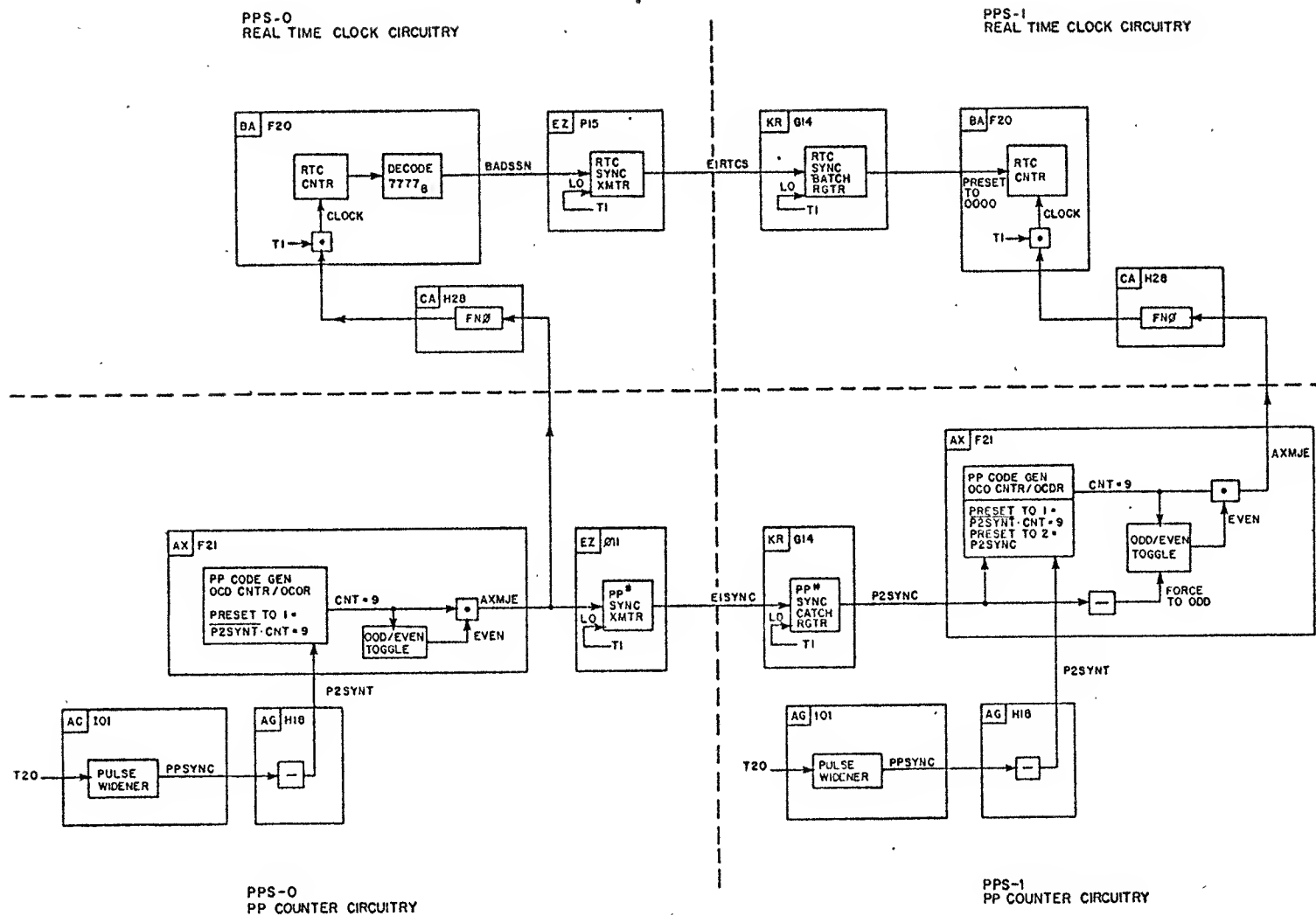


Figure 4-2-8. Inter-PPS Counter Synchronization

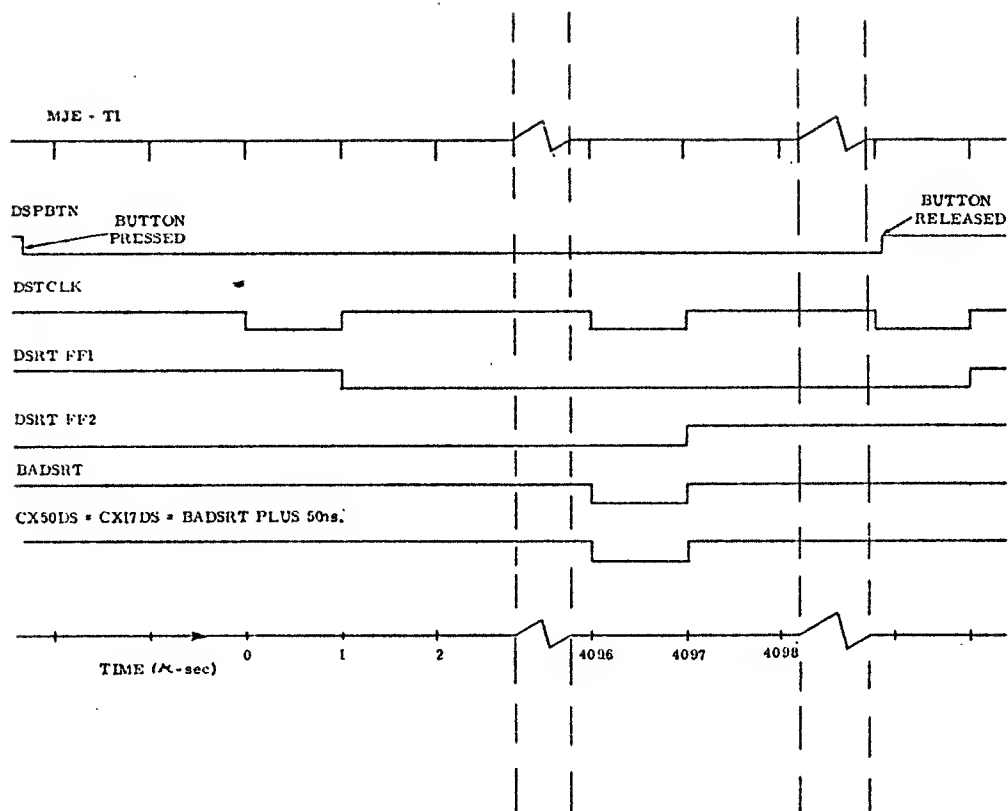


Figure 4-2-9. Dead Start Timing

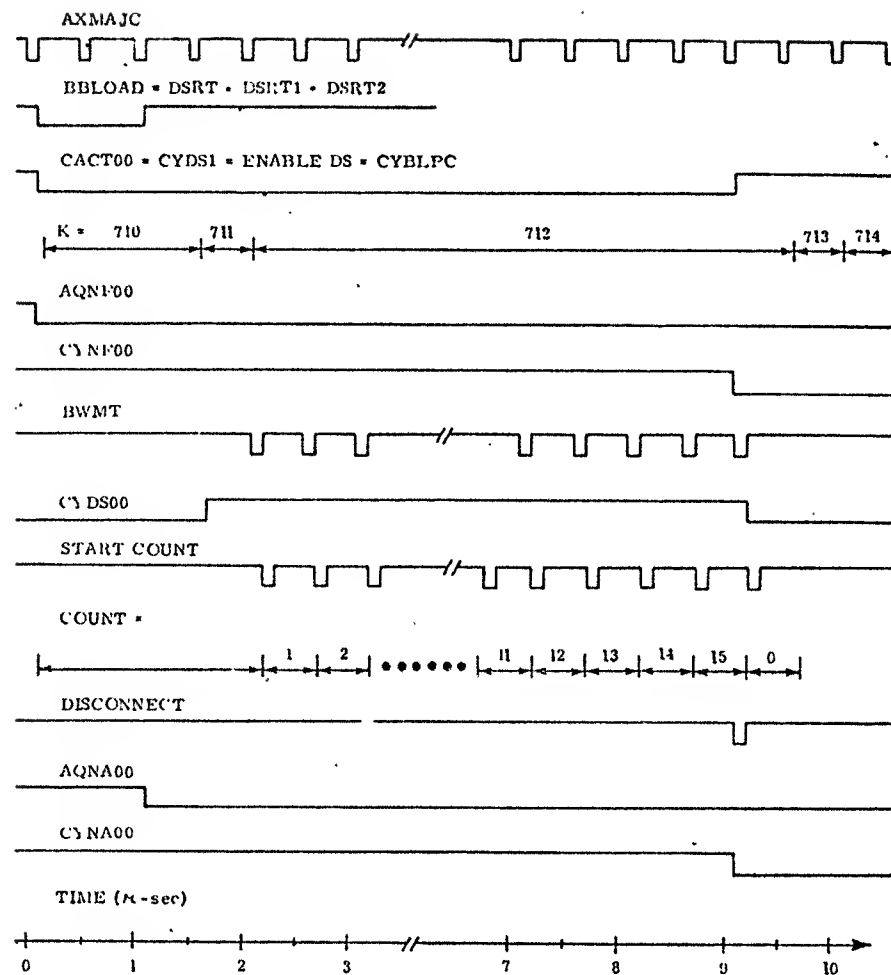


Figure 4-2-10. Loading from Dead Start Panel

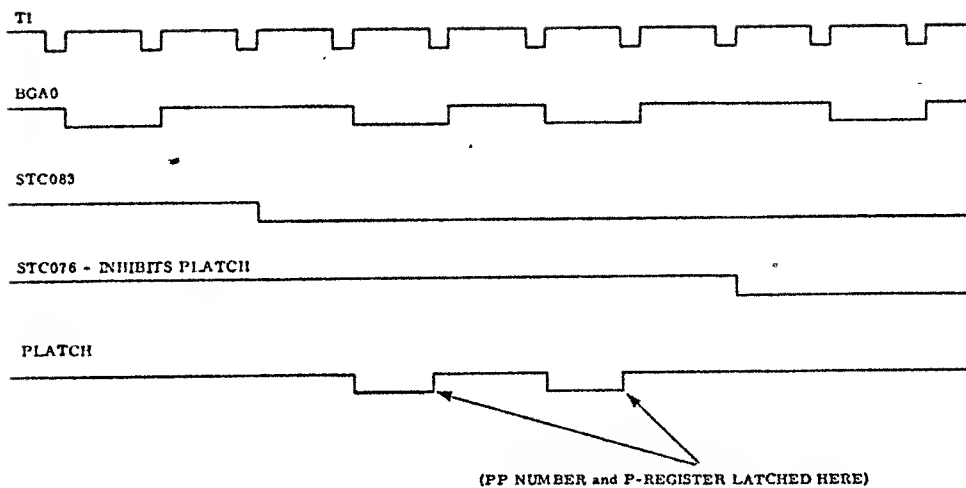


Figure 4-2-11. Breakpoint Mode PP Latch

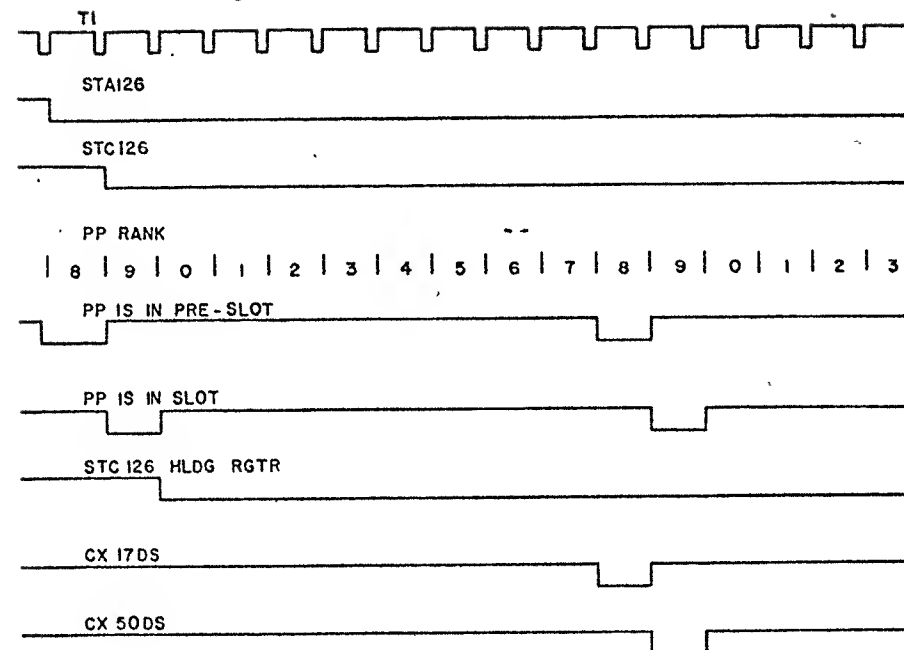
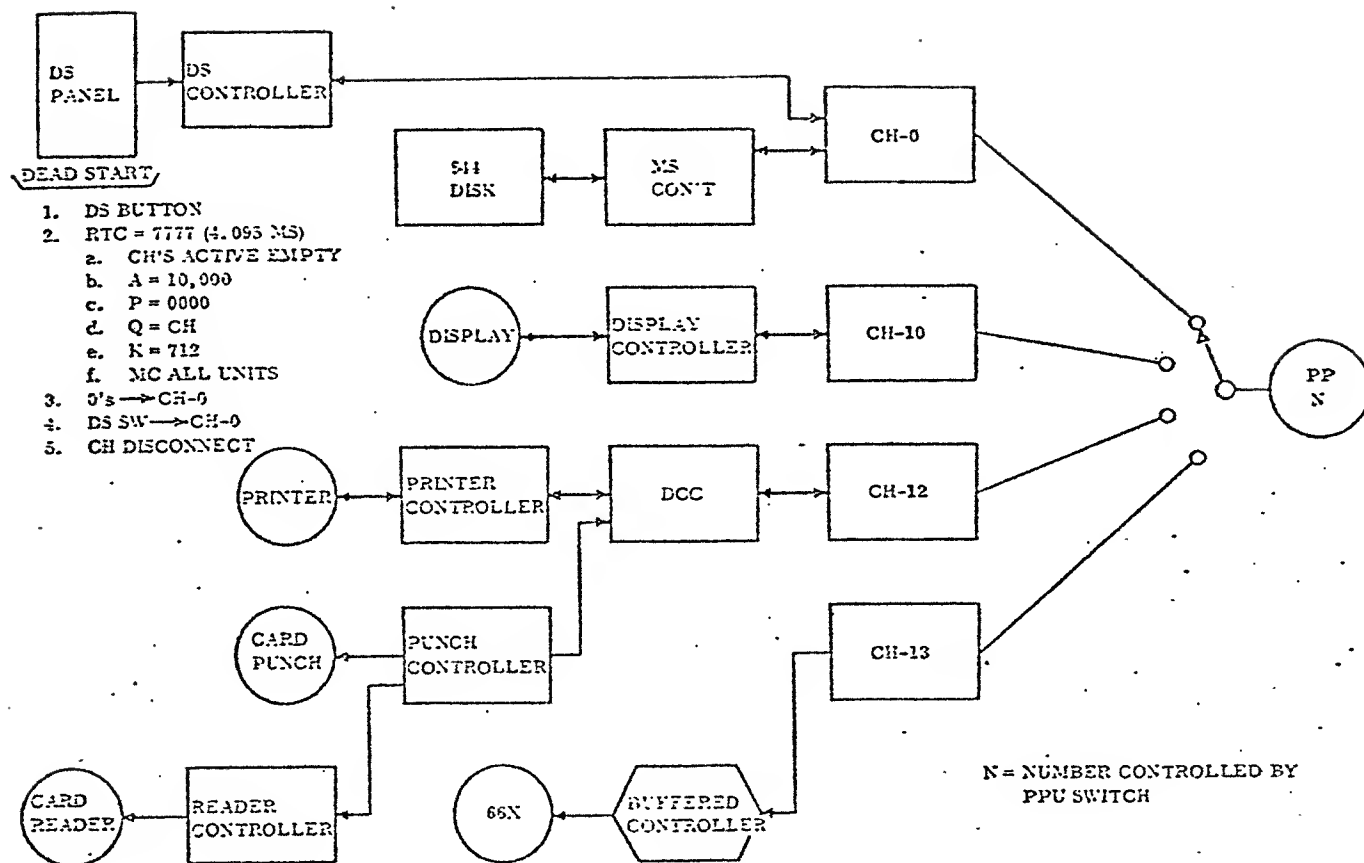


Figure 4-2-12. Single PP Dead Start

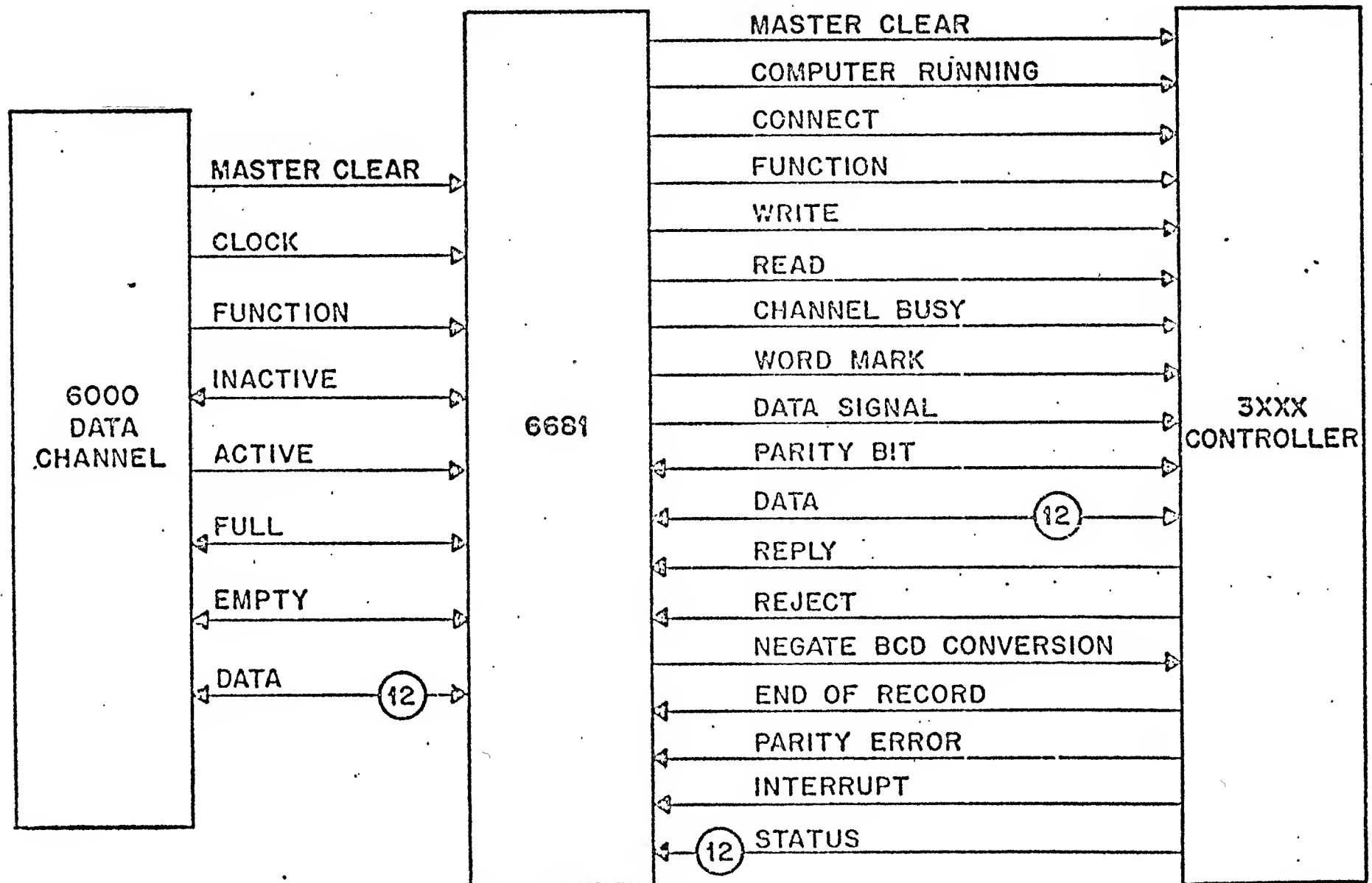


CYBER 170 SIMPLIFIED SYSTEM BLOCK DIAGRAM

A*	B*	C*
DS SETTING FOR TAPE	DS SETTING FOR HELP DISPLAY	DS SETTING FOR CHAIN TEST
1= 7 5 1 3	1= 7 5 1 0	1= 3605
2= 7 7 1 3	2= 7 7 1 0	2= 3607
3= E 0 0 U	3= 7 0 0 2	3= 3611
4= 7 7 1 3	4= 7 4 1 0	4= 1400
5= 0 0 1 0	5= 1 4 0 4	5= 7200
6= 7 7 1 3	6= 7 3 1 0	6= 1414
7= 1 4 0 0	7= 0 0 1 1	7= 7300
10= 7 4 1 3	10= 0 3 0 0	10= 0001
11= 7 1 1 3	11= 6 4 0 0	11= 7500
12= <del>6 6 6 6</del> 0 0 1 3	12= 7 4 0 0	12= 0300
13= <del>0 0 1 0</del>	13= 1 0 4 4	13=
14=	14= 1 4 2 0	14=
15=	15=	15=
16=	16=	16=
17=	17=	17=
20=	20=	20=

\*FUNCTION CODES TO BE REVISED FOR 66

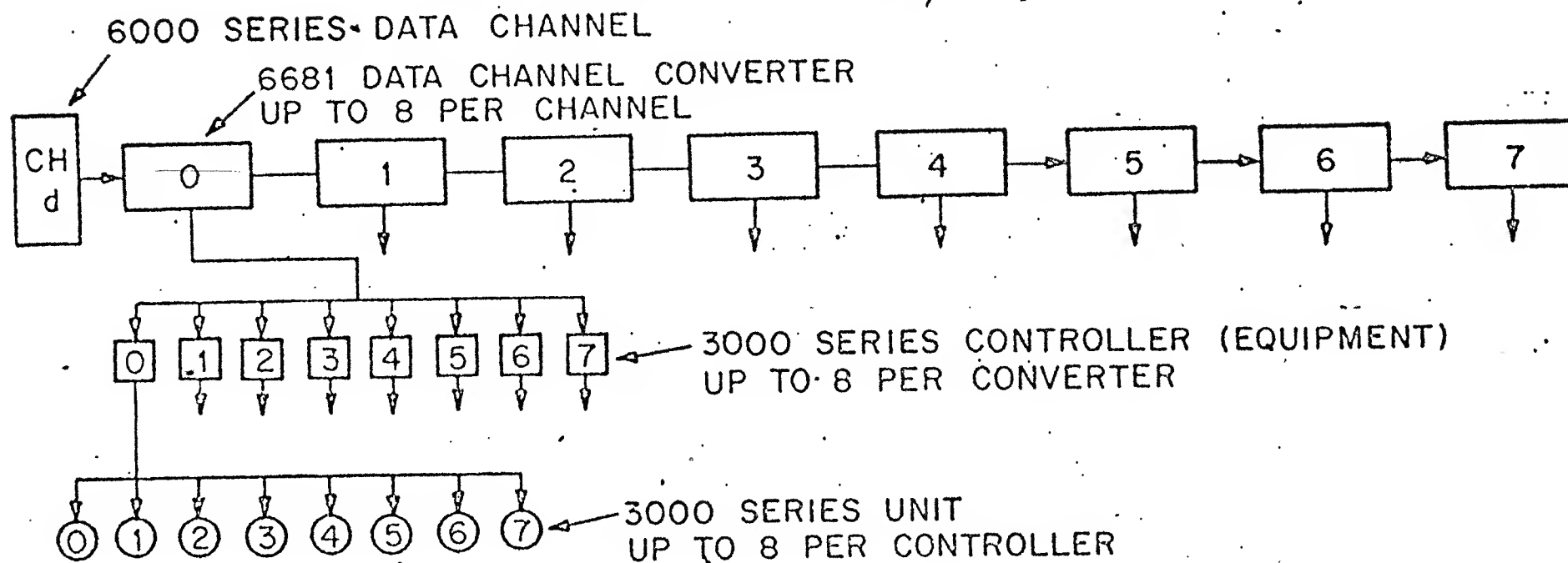
# 6681 COMMUNICATION



# COMPUTER INPUT/OUTPUT COMPONENTS

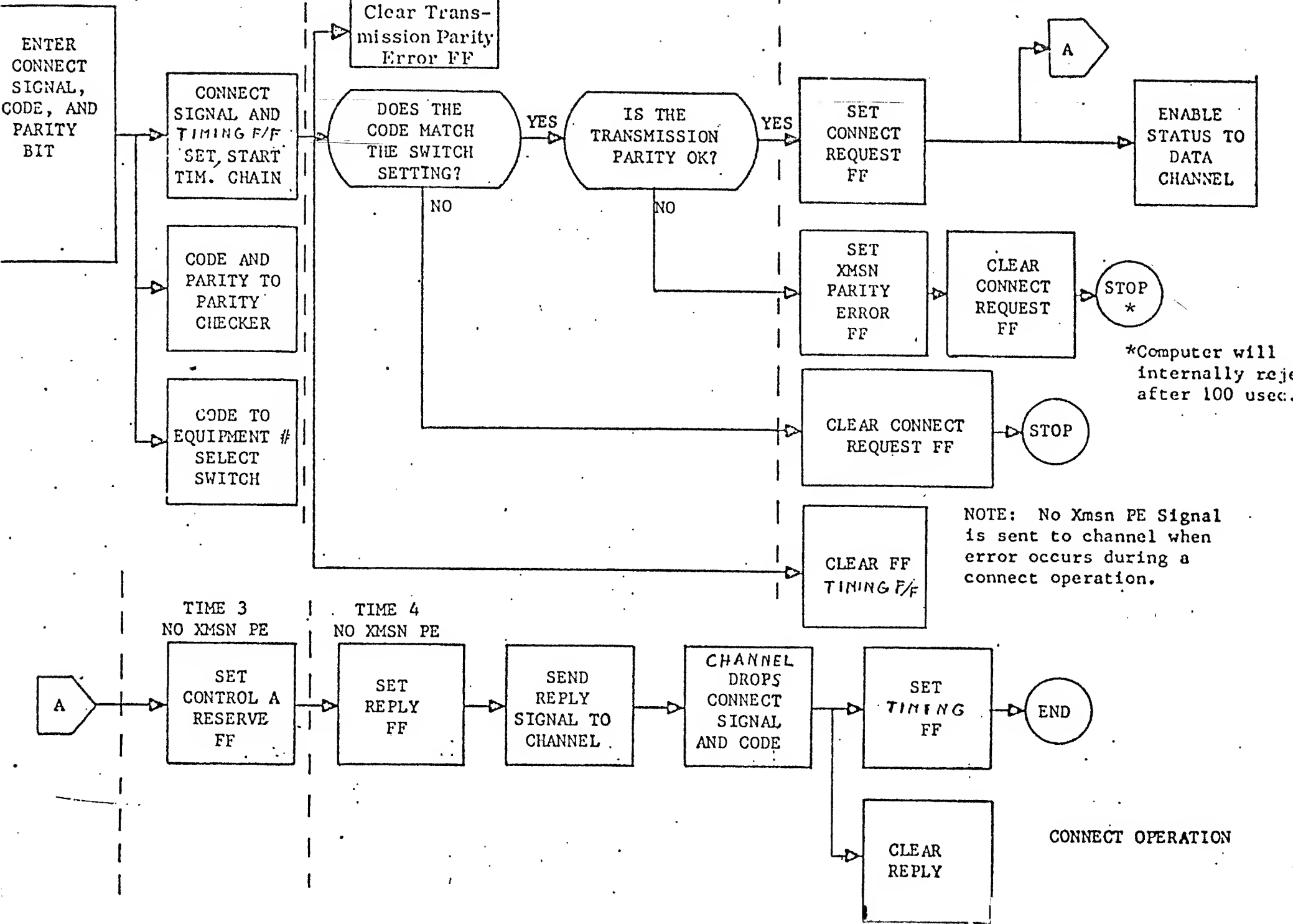


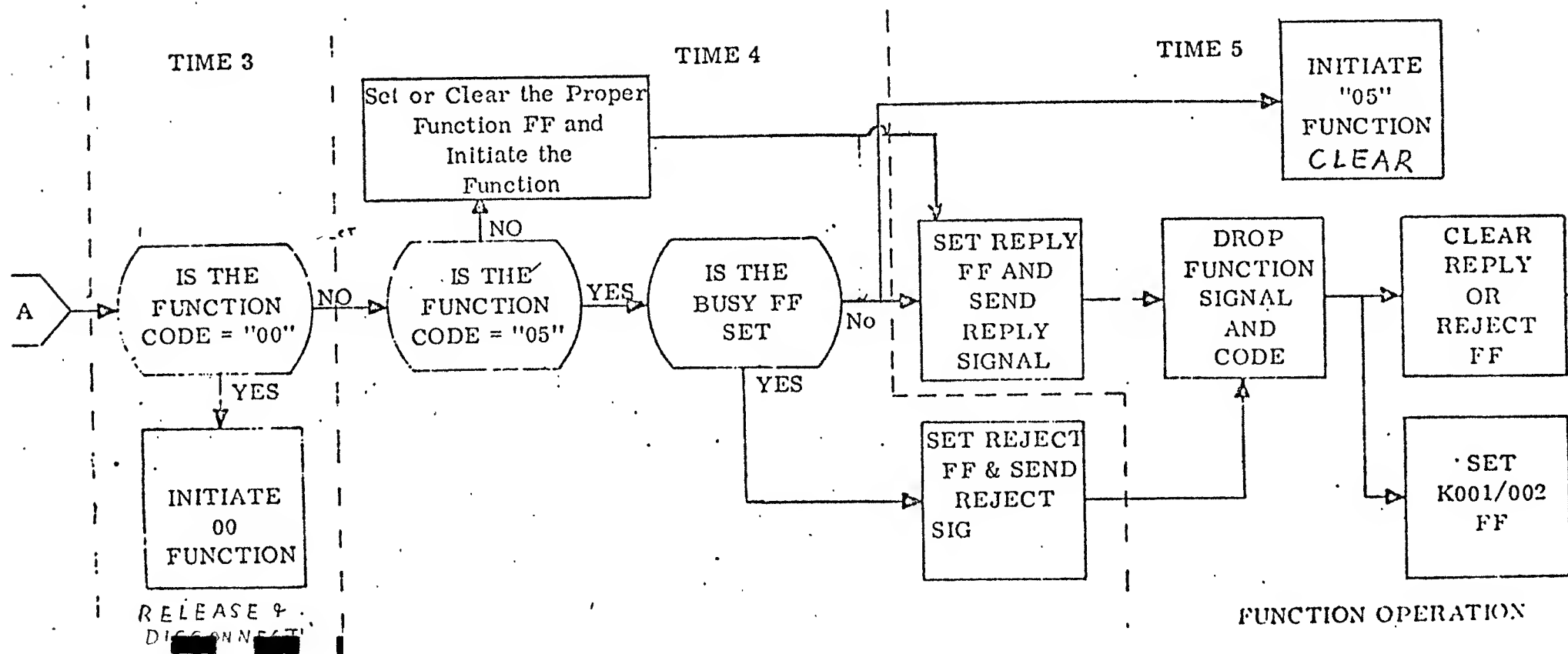
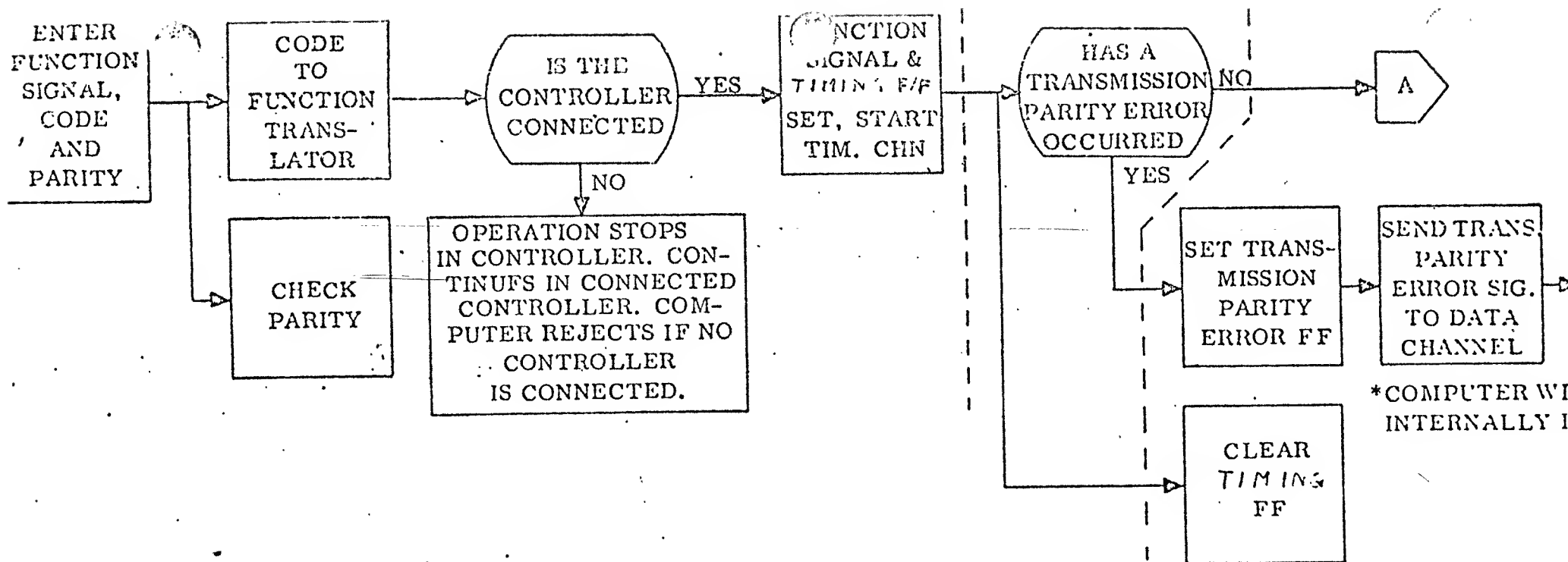
# 6681 DATA CHANNEL CONVERTER COMPONENT RELATIONSHIPS



## 3000 SERIES CONTROLLERS VS. NO. UNIT ATTACHED

MAGNETIC TAPE	8 EA
CARD PUNCH	1 EA
CARD READER	1 EA
LINE PRINTER	1 EA
DRUM	1 EA
MASS STORAGE (DISC)	8 EA





# SIGNALS FROM CHANNEL TO EXTERNAL EQUIPMENT

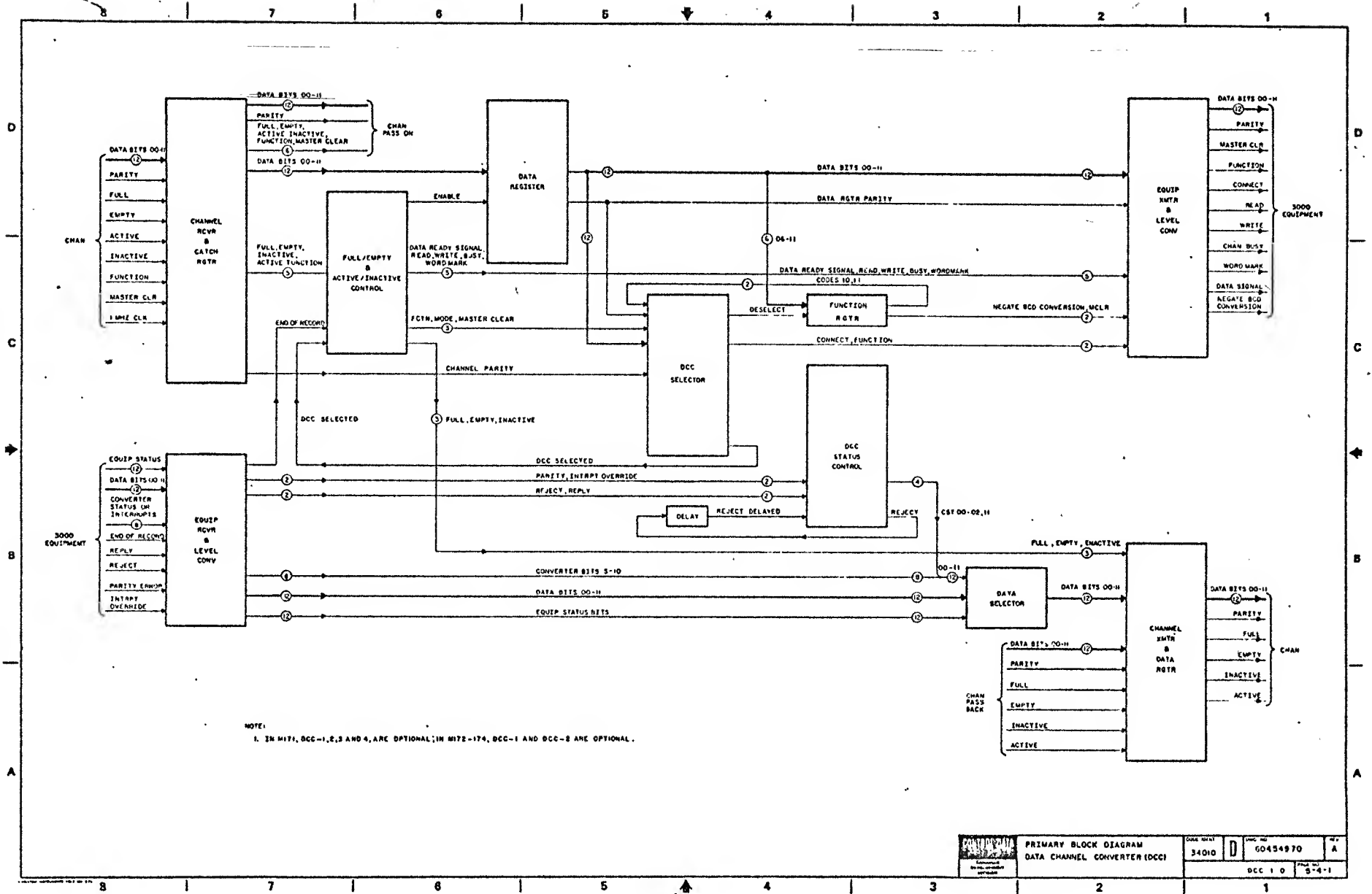
Signal	Definition	Signal	Definition
Connect	<p>"1" signal sent to external equipment when 12-bit Connect code is available on data lines. If the equipment is available, it connects and returns the Reply signal. If it is not available, it returns the Reject signal. The Connect signal and code drop when the external equipment returns the Reply or Reject.</p> <p>A channel may have a maximum of eight external equipments attached to its set of I/O cables. The 12-bit Connect code and the Connect signal are received by all eight equipments, but only one equipment (the Equipment Number switch setting of which corresponds to the upper 3 bits of the Connect code) responds. The other equipments do not return Reject signals.</p> <p>No response is returned by any of the equipments if a parity error exists on the Connect code; however, the Transmission Parity Error indicators on all equipments light. After a delay of 100 microseconds, the I/O channel generates its own internal Reject.</p> <p>A Connect code does not initiate any action, but merely selects an external equipment. The upper 3 bits of the Connect code select one of the eight possible equipments attached to the channel, and the lower 9 bits specify the unit of that equipment.</p> <p>A Connect code matching its Equipment Number switch setting is accepted by the equipment if it is available, although it may be in the Not Ready condition. The Connect code is rejected only if the equipment is already connected to another channel, or is otherwise not available. The equipment does, however, enable its Status lines to the channel which attempted to connect, so that the reason for the Reject may be determined using the Copy Status instruction. The Status lines remain enabled to that channel until it transmits another Connect code to any of its external equipments.</p> <p>Once an equipment is connected to a channel, it remains connected until the program initiates a</p> <p>Master Clear. Any Connect code which does not match its Equipment Number switch setting disconnects the equipment, including its status lines. The equipment must be capable of recognizing the code and disconnecting within 1</p>	Connect (Cont'd)	microsecond after receiving the Connect signal. When being selected, the equipment must not return a Reply or Reject earlier than 2 microseconds after receiving the Connect signal.
		Function	<p>"1" signal sent to external equipment when 12-bit Function code is available on data lines. If the connected equipment is capable of executing the specified function at the time that it receives the Function signal, it initiates the function and returns the Reply signal. If the equipment cannot perform the function, it returns the Reject signal. The Function signal and code drop when the external equipment returns the Reply or Reject.</p> <p>The 12-bit Function code and Function signal are received by all the equipments attached to the channel, but only the connected equipment responds. If no equipment is connected, the Function signal and code are completely ignored. After a delay of 100 microseconds the I/O channel generates its own internal Reject.</p> <p>The specified function is not performed if a parity error exists on the Function code; however, a Parity Error signal is returned by the connected equipment. Also, the Transmission Parity Error light on the connected equipment lights.</p> <p>Once a Function code is accepted, all other function codes are locked out until the first one is acted upon. An equipment does not hold or stack the function codes; a Reply or Reject is returned immediately. If a second Function code is received which specifies the same function as the previous Function code, the second Function code is rejected unless the function can immediately be performed a second time.</p>
		Channel Busy	Static "1" signal sent to external equipment while the channel is active during a Read or Write operation. The Channel Busy signal becomes a "1" immediately after the computation module initiates the Read or Write operation and remains up until the operation is finished. It does not drop between blocks of data in a chain, but does drop when an End of Record signal is returned by the external equipment, as this signal terminates the current Read operation.

SIGNALS FROM CHANNEL TO EXTERNAL EQUIPMENT (Cont'd)

Signal	Definition	Signal	Definition
Channel Busy (Cont'd)	The connected equipment immediately becomes Busy (status response bit 01) when the Channel Busy line goes to a "1", unless it is already Busy or is Not Ready. If the equipment is already Busy finishing a previous operation, it remains Busy and begins the new operation as soon as possible. The equipment does not become Busy if it is Not Ready. However, if the equipment becomes Ready while the Channel Busy signal is up, the equipment becomes Busy.	Word Mark	A "1" signal sent to external equipment together with certain of the Data signals of both Read and Write operations. The Word Mark signal comes up 0.1 microseconds in advance of the Data signal, and remains up for the duration of the Data signal.
Read	Static "1" signal directing the connected equipment to begin reading information from its storage medium, and to continue as long as the Read signal is present. The Read operation always starts at the beginning of a record. If the Read signal drops before the complete record is read, data transmission stops but the external equipment continues its action until the End of Record is reached. If the Read signal drops and comes back up with a record, data transmission stops and does not begin again until the beginning of the next record.	Master Clear	"1" signal from 6681 which returns the channel and external equipment to zero initial conditions and disconnects external equipment.
Write	Static "1" signal directing the connected equipment to begin writing information into its storage medium, and to continue writing as long as the Write signal is present. The Write operation always starts at the beginning of a record. Each time the Write signal drops, the external equipment automatically ends the record.	Computer Running	Static "1"
Data Signal	"1" signal used during Read and Write operations. Data signal drops when Reply (or End of Record) is received from external equipment. 1) During a Read operation, the Data signal indicates that the channel is ready to accept data from the external equipment. 2) During a Write operation, the Data signal indicates that the channel has placed output data on the data lines.	Suppress Assembly/Disassembly	A "1" signal that disables the Assembly/Disassembly logic in the external equipment.  Any equipment or controller which does not normally perform 12-bit/6-bit Assembly/Disassembly ignores the Suppress Assembly/Disassembly signal.

# SIGNALS FROM EXTERNAL EQUIPMENT TO CHANNEL

Signal	Definition	Signal	Definition
Reply	<p>"1" signal produced by external equipment in response to a Connect, Function, or Data signal. Signal drops when Connect, Function, or Data signal drops.</p> <ol style="list-style-type: none"> <li>1) If connection can be made when Connect signal is received, external equipment connects and returns a Reply.</li> <li>2) If specified function can be performed when Function signal is received, external equipment initiates function and returns a Reply.</li> <li>3) During a Read operation, external equipment sends a Reply as soon as it has placed data on the data lines in response to the Data signal. During the Write operation, external equipment sends a Reply as soon as it samples the data lines in response to the Data signal.</li> </ol> <p>(If End of Record is reached during a Read operation, the Reply is not returned in response to the Data signal. Instead, the external equipment transmits the End of Record signal.)</p>	Parity Error (cont'd)	<ol style="list-style-type: none"> <li>c) The Transmission Parity Error lights on all equipments attached to the channel comes on.</li> <li>2) Parity Error on Function code <ol style="list-style-type: none"> <li>a) Nothing happens if no equipment is connected.</li> <li>b) If an equipment is connected, the following occurs: <ol style="list-style-type: none"> <li>1) It returns a Parity Error signal.</li> <li>2) The Transmission Parity Error light comes on.</li> <li>3) It does not return a Reply or Reject.</li> <li>4) It does not perform the function.</li> </ol> </li> </ol> </li> <li>3) Parity Error on data during Write operation. <ol style="list-style-type: none"> <li>a) Nothing happens if no equipment is connected.</li> <li>b) If an equipment is connected, the following occurs: <ol style="list-style-type: none"> <li>1) It returns a Parity Error signal.</li> <li>2) The Transmission Parity Error light comes on.</li> <li>3) It uses the data.</li> <li>4) It returns a Reply.</li> </ol> </li> </ol> </li> </ol>
Reject	"1" signal produced by external equipment in response to a Connect or Function signal, if the connection cannot be made or the function cannot be performed at the time that the external equipment receives the respective signal.	Status Bits	The external equipment indicates its operating conditions by placing information on the 12 status lines. Each equipment has its own particular set of Status Response codes. Refer to 3000 Series Peripheral Equipment Reference Manual (Pub. No. 60108800) for definitions of all External Status codes.
End of Record	<p>"1" signal produced (instead of Reply) in response to the next Data signal following the end of every record during a Read operation. The End of Record signal drops when the Data signal drops.</p> <p>If the Read signal drops before the End of Record is reached, the End of Record signal is not sent, although the external equipment continues its action until the end of record is reached.</p>	Interrupt Lines	<p>A "1" signal on an Interrupt Line indicates that an external equipment has reached a predetermined condition. A channel may communicate with a maximum of eight equipments and each equipment uses one Interrupt line.</p> <p>Each equipment has a set of conditions upon which it interrupts if selected. Refer to the 3000 Series Peripheral Equipment Reference Manual (Pub. No. 60108800) for a list of interruptible conditions for each external equipment.</p>
Parity Error	<p>"1" signal produced if the channel does not send an odd number of "1's" in each 12 bits plus parity bit. (A parity bit accompanies each 12 bits)</p> <p>The following events occur when a Parity Error is detected.</p> <ol style="list-style-type: none"> <li>1) Parity Error on Connect code <ol style="list-style-type: none"> <li>a) No equipment connects.</li> <li>b) Any connected equipment disconnects.</li> <li>c) No equipment returns a Reply or Reject.</li> <li>d) No equipment returns a Parity Error signal.</li> </ol> </li> </ol>		



# PRIMARY BLOCK DIAGRAM (DCC 1, 0)

The data channel converter (DCC) is an interface required to allow the use of 3000 series equipment on CYBER 170 series computer systems. There are two data channel converters in the peripheral processor subsystem (PPS) chassis.

The DCC attaches to one of the data channels of the PPS and may share that channel with a maximum of seven other CYBER 170 series peripheral equipments. All equipment attached to a channel is connected in a serial manner; the first equipment on the channel must pass data on to the second, which must pass it on to the third, and so on. Data must be passed back along the chain in a similar manner. At each piece of equipment, the data is resynchronized at 100-ns intervals. When the DCC is selected, the data pass-on is continued. However, nonselected equipment ignores information so received.

The output signals from a data channel consist of a 12-bit data word and a parity bit, and the following control signal lines:

FULL	- indicates that a 12-bit word is in the device's output register awaiting processing.
EMPTY	- indicates that the data accompanying a full signal has been received and processed.
INACTIVE	- indicates that <u>functions</u> can be processed.
ACTIVE	- indicates that <u>data</u> can be processed.
FUNCTION	- accompanies a 12-bit word to indicate that it is to be interpreted as a function code; i.e., to perform an operation within an equipment.

When a DCC is selected, it must convert data signals received at its channel input into signals suitable to operate 3000 series equipment and also translate responses from that equipment. In addition to the 12-bit data words, the following signals are exchanged:

1. CONNECT - This signal accompanies a connect code on the data lines. The connect code addresses a 3000 equipment.
2. FUNCTION - Accompanies a function code on the data lines. Signifies an operation to be performed.

3. WRITE - This signal remains up during a data transfer to the 3000 equipment.
4. READ - This signal remains up during a data transfer from the 3000 equipment.
5. BUSY - Generated by a read or write. This signal remains up during a data transfer.
6. WORD MARK - Accompanies every 12-bit data word transferred to the 3000 equipment.
7. DATA SIGNAL - Accompanies word mark, but delayed by 0.1  $\mu$ sec.
8. REPLY - Positive response by 3000 equipment, acknowledging the receipt of data, connect or function codes.
9. REJECT - Negative response by 3000 equipment to data, connect or function codes.
10. END OF RECORD - Generated in the 3000 equipment to indicate a break in a read data flow; e.g., record gap on tape.
11. INTERRUPT - Signal sent by 3000 equipment when a predetermined condition has been reached. (Refer to individual equipment manuals.)
12. STATUS - 12 bits of data forming a code to indicate the equipment status. Sent from 3000 equipment on request.
13. PARITY BIT - Accompanies the 12 bits on the data lines during a transfer to the 3000 equipment. Brings the total number of logical ones on the 13 lines to an odd number (odd parity).
14. PARITY ERROR - Raised by the 3000 equipment when its calculation of parity does not agree with condition of the parity bit.
15. MASTER CLEAR - Returns all 3000 equipment to initial operating conditions.
16. NEGATE BCD CONVERSION - Prevents conversion from internal to external BCD. Normally up for CYBER 170.

Two sets of codes are required to operate a 3000 series equipment via a DCC:

- (a) Function and status response codes for the DCC.
- (b) Connect, function and status codes for a specific 3000 series equipment.

The converter codes (table 4-2-10) allow the computer system to connect 3000 series equipment, and to transmit function codes to that equipment. In addition, the codes allow the sensing of conditions in both the converter and the 3000 series equipment, and the interchange of data between the channel and equipment.

#### FUNCTION CODES (CHANNEL TO DCC)

##### 2000 Select Converter

This code selects the DCC from the equipment sharing the data channel.

##### 2100 Deselect Converter

This code deselects the DCC. This must be done before other equipment on the data channel can be selected.

##### Note:

When two DCCs share a data channel, one converter is assigned different select and deselect codes (e.g., 2200 and 2300, or 2400 and 2500).

##### NUUU Connect Equipment Mode I

Connects equipment 4, 5, 6 or 7 units UUU. N = equipment number (restricted to 4 through 7); UUU = unit number.

##### 1000 Connect Initiate Mode II

Specifies mode II operation and directs the DCC to send the next data word received to the 3000 series equipment as a connect code. The next data word will be in the form NUUU, where N = equipment number (1 through 7), and UUU = unit number.

##### 0FFF Function Transmit Mode I

Specifies mode I operation and causes the DCC to transmit this 12-bit function code (0FFF) to the connected equipment. FFF can be the lower 9 bits of any 12-bit code whose upper 3 bits are zeros.

TABLE 4-2-10. DCC CODES

<u>Select/Deselect</u>	
Select Converter	2000*
Deselect Converter	2100
<u>Connect</u>	
Connect Equipment (Mode I)	NUUU**
Connect Equipment (Mode II)	1000
<u>Function</u>	
Function Transmit (Mode I)	0FFF#
Function Initiate (Mode II)	1100
<u>Data Input/Output</u>	
Input EOR Initiate	14XX <sup>n</sup>
Input Initiate	15XX
Output Initiate	16XX
<u>Master Clear</u>	
Function Master Clear	1700
<u>Status Request</u>	
Converter Status Request	1200
Equipment Status Request	1300
<p>* Where two converters share a common data channel, one of the converters is assigned different select and deselect codes, such as 2200 and 2300, or 2400 and 2500.</p> <p>** N = 4-7 (equipment number). UUU = lower 9 bits of connect code.</p> <p># FFF = lower 9 bits of function code.</p> <p><sup>n</sup> Initiate conditions are defined by XX.</p>	

#### 1100 Function Initiate Mode II

Specifies mode II operation; directs the DCC to transmit the next data word received to the connected equipment as a function code.

#### 14XX Input EOR Initiate

Prepares the DCC for a 3000 series read operation. It permits the termination of the read by either an end-of-record signal from the equipment or by a channel disconnect.

#### 15XX Input Initiate

Prepares the DCC for a 3000 series read operation. It permits the termination of read only by a channel disconnect, and should not be used for magnetic tape inputs.

#### 16XX Output Initiate

Prepares the DCC for a channel output operation and permits termination by a channel disconnect.

##### Note:

The 14XX, 15XX and 16XX function codes use the XX portion to define the initiate conditions. The least significant bit of the lowest octal digit defines the BCD conversion condition. A logical one negates BCD conversion. This condition remains in effect until a 14XX, 15XX or 16XX code is received in which the LSB is a logical zero.

#### XX4X, XX6X Deactivate

These are options that enable two additional methods of generating an inactive signal in the converter during a read or write operation. The codes are combined with the input/output functions.

#### 146X, 156X, 166X

Enable the DCC to send an inactive signal to the channel when an interrupt override signal is generated in the 3000 series equipment. The interrupt override signal is generated in the 3000 series equipment when interrupt on an abnormal end-of-operation is selected and an abnormal condition exists. This code may be used for any equipment that has an interrupt override feature.

Power Failure Note: In model B the above applies but, additionally, a power failure will generate the inactive signal when the XX6X codes are in effect.

#### 144X, 154X, 164X

Used when the 3000 series equipment does not have an interrupt override feature. When an abnormal end-of-operation occurs, the abnormal end-of-operation status (1XXX) is returned to the DCC. The DCC senses this condition and generates an inactive signal which is transmitted to the channel.

Power Failure Note: In model B the above applies but, additionally, a power failure will generate the inactive signal when the XX4X codes are in effect.

#### 1700 Function Master Clear

This code master clears the DCC and conditions within it.

#### 1200 Converter Status Request

Permits the DCC to input the converter status. A one-word input must follow to read in the status response.

#### 1300 Equipment Status Request

Permits the DCC to input the status response from the connected 3000 series equipment. A one-word input must follow to read in the status word.

##### Note:

Any 1XXX function code sent to the DCC clears any previous 1XXX function condition.

#### Status Reply Codes

There are two types of status reply codes available to the channel from the DCC:

##### 1. Converter Status

This code indicates the status of all the 3000 equipment attached to it. It comprises the reply/reject state, the parity error condition and the interrupt signal state from each attached equipment. (see table 4-2-11.)

##### 2. Equipment Status

This 12-bit code is generated within a 3000 series equipment. It is available when the equipment is connected to the DCC and when the equipment rejects a connect code. Each bit in the equipment request code indicates a condition within the equipment. Codes vary with equipment type and are listed in the individual manuals.

TABLE 4-2-11. CONVERTER STATUS REPLY CODES

CODE	DESCRIPTION
XXX0	REPLY
XXX1	REJECT (INT OR EXT)
XXX2	REJECT (INT)
XXX4	TRANSMISSION PARITY ERROR. DETECTED BY EITHER THE CONVERTER OR THE 3000 SERIES EQUIPMENT. IN MODEL B THIS CODE ALSO ARISES ON POWER FAILURE.
XX1X to 2XXX	EQUIPMENT INTERRUPTS
4XXX	TRANSMISSION PARITY ERROR ON DATA FROM CHANNEL

DEFINITION OF CONVERTER STATUS CODESXXX0 Reply

Indicates acceptance by the 3000 series equipment of a connect or function code.

XXX1 Reject (Int or Ext)

The least significant bit of the lower octal digit is set when the 3000 series equipment returns a reject signal to the DCC in response to a connect or function code. This bit is also set when the DCC generates an internal reject.

XXX3 internal Reject

Bits 0 and 1 are set after a delay of 100  $\mu$ sec if the 3000 series equipment fails to return a reply or a reject in response to a connect or function code.

XXX4 Parity Error

Bit 2 is set when the 3000 series equipment detects a parity error, or if the DCC senses a parity error on data received from the 3000 series equipment. It is also set when, in mode II operation, the DCC detects parity error in function codes or data from channel.

XX1X to 2XXX Equipment Interrupts

All 3000 series equipments have an interrupt feature that enables them to notify the DCC when specific operating conditions occur. Most of these use an interrupt which can be selected or released by function codes. The reference manual for each 3000 series equipment gives interrupt selection codes and defines interrupt conditions. When a selected interrupt occurs, the equipment sends a signal to the DCC by setting a bit in the converter status word. Bits 3 through 10 of the status word indicate interrupts on any one of the eight possible equipments attached to the DCC. The bit and equipment relationship is detailed in table 4-2-12.

TABLE 4-2-12. CSB/EQUIPMENT RELATIONSHIP

EQUIPMENT NUMBER	0	1	2	3	4	5	6	7
CONVERTER STATUS BIT	3	4	5	6	7	8	9	10

An equipment can send an interrupt to the DCC whether connected or not. This provides a limited status check on each 3000 equipment.

The interrupt status bit remains set in the DCC as long as the equipment maintains the signal. An interrupt signal can be cleared by any one of the following:

- (a) A converter master clear (1700) that clears all the 3000 series equipments attached to the DCC and the DCC itself.
- (b) A function code sent to the specific interrupting equipment.
- (c) A dead start master clear from the channel.

4XXX Transmission Parity Error on Channel Data

Bit 11 is set when the equipment detects a parity error on data received from channel. At the same time, bit 2 is also set.

## OPERATION

After the DCC has been selected, any one of a maximum of eight equipments attached to it can be connected. The connect operation activates that equipment and deactivates all others.

Each 3000 series equipment is assigned a number (0 through 7) by means of its 8 position equipment number switch. Each equipment may have subordinate units attached to it, and each of these is assigned a unit number. To connect the equipment, a 12-bit connect code is required. (See figure 4-2-17 for format.)

BIT NUMBER											
11	10	9	8	7	6	5	4	3	2	1	0
N			U			U			U		
EQUIP NO.			UNIT SELECT NO.								

Figure 4-2-17. Connect Code Format

### Note:

Equipments with no subordinate units have bits 0 through 8 set to zero in their code.

## Connect Codes

A connect code is sent from the channel to the equipment via the DCC. Two methods of doing this are used:

### Mode I Connect

Requires only one DCC function code from the channel, but is restricted to connecting equipments 4 through 7.

### Mode II Connect

Requires a DCC function code followed by a one-word data code. It can connect any of the eight possible equipments.

Once established, a connect is broken only by connecting another equipment, using a dead start master clear or a function master clear (1700). Deselecting the converter does not clear a connect.

## Details of Mode I Connect

Whenever the channel sends a function code in the form 4UUU through 7UUU, a mode I connect operation is performed. The DCC passes the code to the 3000 series equipment as a connect code. Normally, the equipment corresponding to the upper octal digit (4 through 7) connects, and previously connected equipment disconnects. The equipment returns a reply signal to the DCC which, in turn, sends an inactive signal to the channel, thereby making it available for another operation.

If the equipment fails to connect, it returns a reject signal to the DCC. The reject signal causes the DCC to send an inactive signal to the channel and, at the same time, sets status bit 0 in the DCC. This indicates an external reject.

If neither a reply nor reject is returned in response to a connect code, the DCC will generate (after a 100- $\mu$ sec delay) an internal reject, thereby setting status bits 0 and 1 and sending an inactive signal to the channel. The 3000 series equipments check for parity each connect code received from the DCC. If a parity error is detected, no equipment connects and neither reply nor reject is generated; this produces an internal reject condition in the DCC.

## Details of Mode II Connect

- Function code 1000 (connect initiate) is sent to the DCC. This sets conditions in the DCC and no output is made to the 3000 series equipment. The DCC returns an inactive signal to the channel.
- The channel outputs one word containing the connect code. This is forwarded by the DCC to the 3000 equipment.

The three possible responses are the same as in mode I, and any of these causes the DCC to send an empty signal to the channel. After a mode II connect, the converter status response should be checked for a reject.

## Function Codes

After a 3000 series equipment is connected, it accepts 12-bit function codes from the DCC. These establish operating conditions or initiate operations within the equipment. These function codes are distinct from those received by the DCC from the channel. As in the case of connect codes, there are two modes of communication between channel and 3000 equipment:

### Mode I Function

Requires only one channel function instruction, but is restricted to a 9-bit code.

### Mode II Function

Requires a function instruction from the channel to be followed by a one-word data output. In this way, a full 12-bit function code can be sent to the 3000 series equipment.

#### Details of Mode I Function

Similarly to mode I connect, a mode I function operation is performed by the DCC whenever it receives an 0FFF function code from the channel. FFF can be any 9-bit 3000 series function code. The DCC forwards this code to the connected 3000 series equipment as a function code. The 3000 series equipment will return one of two possible responses:

- (a) Reply - indicating acceptance of the code.
- (b) Reject - indicating nonacceptance. This signal sets the external reject status bit in the DCC.

In the event of no response, the DCC will generate an internal reject after a delay of 100  $\mu$ sec. After a mode I function operation, a status check should be carried out to test for a reject or parity error.

#### Details of Mode II Function

As in a mode II connect operation, two steps are necessary:

- (a) Function code 1100 (function initiate) is sent to the DCC from the channel. This code conditions the DCC for a mode II function operation and is not transmitted to the 3000 series equipment. The DCC responds with an inactive signal to the channel.
- (b) A one-word output comprising the desired 12-bit function word is sent to the DCC from the channel. The DCC forwards this to the 3000 series equipment as a function code.

The responses by the 3000 series equipment are the same as for a mode I function. Any or no response causes the DCC to send an empty signal to the channel, indicating receipt of the output word. The results of a parity error at the 3000 series equipment during a mode II function operation are:

- (1) Parity error status bit (2) is set.

(2) After 100  $\mu$ sec, the DCC generates an internal reject which sets the reject status bits.

- (3) The equipment does not accept the function code.

After a mode II function operation, a status check should follow to sense for parity error or reject.

### Data Transfer

Data transfer can proceed only after the DCC has been selected and the 3000 series equipment connected.

#### BASIC PRINCIPLES OF OPERATION (Refer to DCC 1.0)

For the DCC to operate at all, it must be selected; if not, the data pass-on and data pass-back are ineffective. The DCC selected signal enables the gating of the pass pulse which passes data from the CYBER 170 output to the 6000 series output port of the DCC. Similarly, signals waiting at the 6000 series input port of the DCC are passed to the CYBER 170 input.

To select the DCC, the channel receiver and catch register must pass the signals full or function to the active/inactive, full/empty control. Either of these will produce an enable signal shifting the data signals to the output of the data register. From this, register bits 9, 10, 11 are passed to the DCC selector where they must be decoded as a 2. At the same time, bit 6 must be a logical zero. These conditions will raise the DCC selected flag which can only be lowered by a function 2100 input code. The DCC selected flag can also be raised by a master clear.

#### To Connect 3000 Series Equipment

After the DCC is selected, it requires a connect code to select the desired equipment. This connect code can be a single code word (mode I) or two consecutive code words (mode II).

In mode I, the code is transferred immediately to the 3000 series equipment complete with the required control signal. When bit 11 of the data word from the channel is a logical one, indicating a code of 4XXX through 7XXX, a mode I connect flag is raised in the DCC selector, the connect line to the 3000 series equipment is activated and the equipment transmitter and level converter is enabled, thereby making the data word available.

In mode II, the DCC is readied so that the next code will be passed to the converter as the connect code.

When the code 1000 arrives at the input to the DCC, it is decoded in the function register and a flag raised (connect initiate). This inhibits the passing of data and the connect signal until full signal is received on the channel input (i.e., one word later). The second word is passed to the 3000 series equipment as a connect code.

#### Response to a Connect Code

There are two possible responses:

- (a) Reply and
- (b) Reject.

A third possibility is no response.

Reply is a positive response signifying acceptance of the code and its availability. Reject is a negative response signifying acceptance of the code but nonavailability. Both responses store bits in the function register as converter status bits 00 and 01. In the event of no response, these registers are set to indicate internal reject after a time delay.

In all cases, the DCC will send an empty signal to the channel.

#### 3000 Series Function Codes

After the DCC is connected to the 3000 series equipment, function codes may be sent from the channel. As in the case of connect codes, these may be one-word or two-word codes and are designated mode I and mode II, respectively.

Mode I working is signified by a code 0FFF where the lower nine bits are a function code and the top code (0) the mode designator. In this case, the three most significant bits 9, 10 and 11 are decoded in the DCC selector, a function signal is raised on the 3000 series input lines and the data is transferred.

In mode II operation, the initial code sent from the channel is 1100. As in the case of a mode II connect code, this is decoded in the function register. A function initiate flag is raised, inhibiting the passing of data until a full signal is received from the channel. This occurs on the next word, which is passed to the 3000 equipment as a 12-bit function code.

#### Response to a Function Code

The responses are similar to those for a connect code; i.e., reply or reject. As in the case of the connect code, the response sets status bits in the function register. No response will enable the internal reject state to be set. In all cases, the DCC will generate an empty signal to the channel.

#### Data Interchange

To exchange data between the channel and 3000 series equipment, it is necessary to condition the DCC for the type of operation required. There are six of these initiate codes: four enabling data to flow from the 3000 series equipment (input initiate) and two enabling data to flow from the channel (output initiate).

#### Input Initiate Codes

##### 1400 Input Initiate

This code allows the operation to be terminated by either a channel disconnect or an end-of-record signal from the 3000 series equipment. When this input initiate code is detected in the function register, a flip-flop is set which enables a gate in the full/empty active/inactive controller. The receipt of end-of-record causes the inactive signal to be raised. The receipt of the 1400 codes also raises the read and busy lines to the 3000 series equipment.

##### 1500 Input Initiate

This code acts as a 1400 except that the send inactive on EOR flag is not set.

#### Note:

Two other codes, 1401 and 1501, act as 1400 and 1500 respectively except that the condition XXX0, which raises the negate BCD conversion line, is not detected and therefore this signal is absent. For all four codes the transmit enable signal from the function register is suppressed, inhibiting transfer of data to the 3000 series equipment.

##### 1600 and 1601 Output Initiate

The difference between these codes is in the condition of the negate BCD conversion line (as in the input initiate codes). The detection of 16XX occurs in the function register. A flip-flop is set which enables a gate in the full/empty, active/inactive controller. This module generates an empty signal on receipt of a reply, and also raises the control lines write and busy to the 3000 series equipment.

## DETAILED PAK DIAGRAM (DCC 3.0)

### OUTPUT DATA TRANSLATOR

The AP paks receive channel data and control signals and convert them to ECL logic levels. This data is latched into catching registers and then passed on by the TPOD signal. The pass-on transmitters reconvert from ECL to channel logic levels. Registers are cleared by the T20 pulse.

The AD (P28) pak has a 12-bit data register which is loaded by the DWEN1 signal (T10 gated by full and function). The outputs of the register are applied to a parity generator to produce the ADIPAR parity bit.

Parity from channel (DCIPAR) is latched into a flip-flop on the DY pak (O31) by the DWEN1 signal. The flip-flop output is applied to one of the inputs of a parity checker which has the ADIPAR signal as its other input. The output is a parity check that produces a signal (DUICHE) upon detection of a difference between the two inputs. This signal is used to prevent the DCC from being selected by a code with a parity error.

Each DS pak has the following logic level change facilities:

- (a) Two ECL to 3000 series level converters.
- (b) Four 3000 series to ECL level converters.
- (c) Two ECL to 3000 series level converters with an enable control. The output lines of these converters are in common with the input lines of (d), as well as being connected to the 3000 series equipment.
- (d) Two 3000 series to ECL level converters with input lines connected to (c).

The common (c) and (d) connections are used for data transfer to and from the 3000 series equipment. Those parts of the DS paks used on this diagram transfer data and control signals (when the DCC is selected) to the 3000 series equipment.

#### Note:

Detailed pak diagram 3.0 covers both DCC-1 and DCC-2.

DETAILED PAK DIAGRAM (DCC 3.1)  
CONTROL SIGNAL TRANSLATOR

The DU (Q19, DCC selector) pak has a decoder for channel bits 9, 10, 11. When a 2 is decoded, the decode is applied to an AND gate enabled by the function flip-flop. The gate output acts as a clock for the pass flip-flop which has bit 6 as its data input. If bit 6 = 0 (code 20XX), the pass FF is cleared (DUSEL). This FF is cleared by master clear. If bit 6 = 1 (code 21XX), the equipment is deselected. When the DCC is selected, the DISLFN signal enables flip-flops on the DV pak to decode the second octal character of codes beginning 1XXX. The detection of bits 9, 10, 11 = 0 raises the function line to the 3000 series equipment.

The DV (Q20, function register) pak has a decoder for channel bits 6, 7, 8 and is used to determine the second octal digit of a function code. When the first digit of the code is a 1, DISLFN enables the flip-flops connected to the decoder outputs. Depending upon the code, one of these outputs will be active and a flip-flop will latch. The decode 17XX (function master clear) is OR'd with the system master clear to produce a reset for all flip-flops. Also on this pak are flip-flops to set the converter status bits 00 and 01 (see DCC 3.2).

The DW (Q23, full/empty; active/inactive control) produces the necessary signals for the channel (full, empty, active) and for the 3000 series equipment (word mark, read, write and busy).

Note: Detailed pak diagram 3.1 covers both DCC-1 and DCC-2.

Model B differences: In model B, hardware is provided to deactivate and deselect the DCCs in the event of a power failure (see DPD 7.1.1). The DCCs are deselected to reduce channel activity, thereby allowing speedier processing of a power failure routine.

Power failure sets SCR bit 36 which, in turn, sets the power failure mode (PFM) FF at all system DCCs (up to 4 in a dual PPS system). If the deactivate on XX4X or XX6X FFs (DY paks - DCC 3.2) are set, this will send an inactive to the channel. Also, the PFM FF deselects the DCC on any function signal and sets converter status bit 02 (code XXX4).

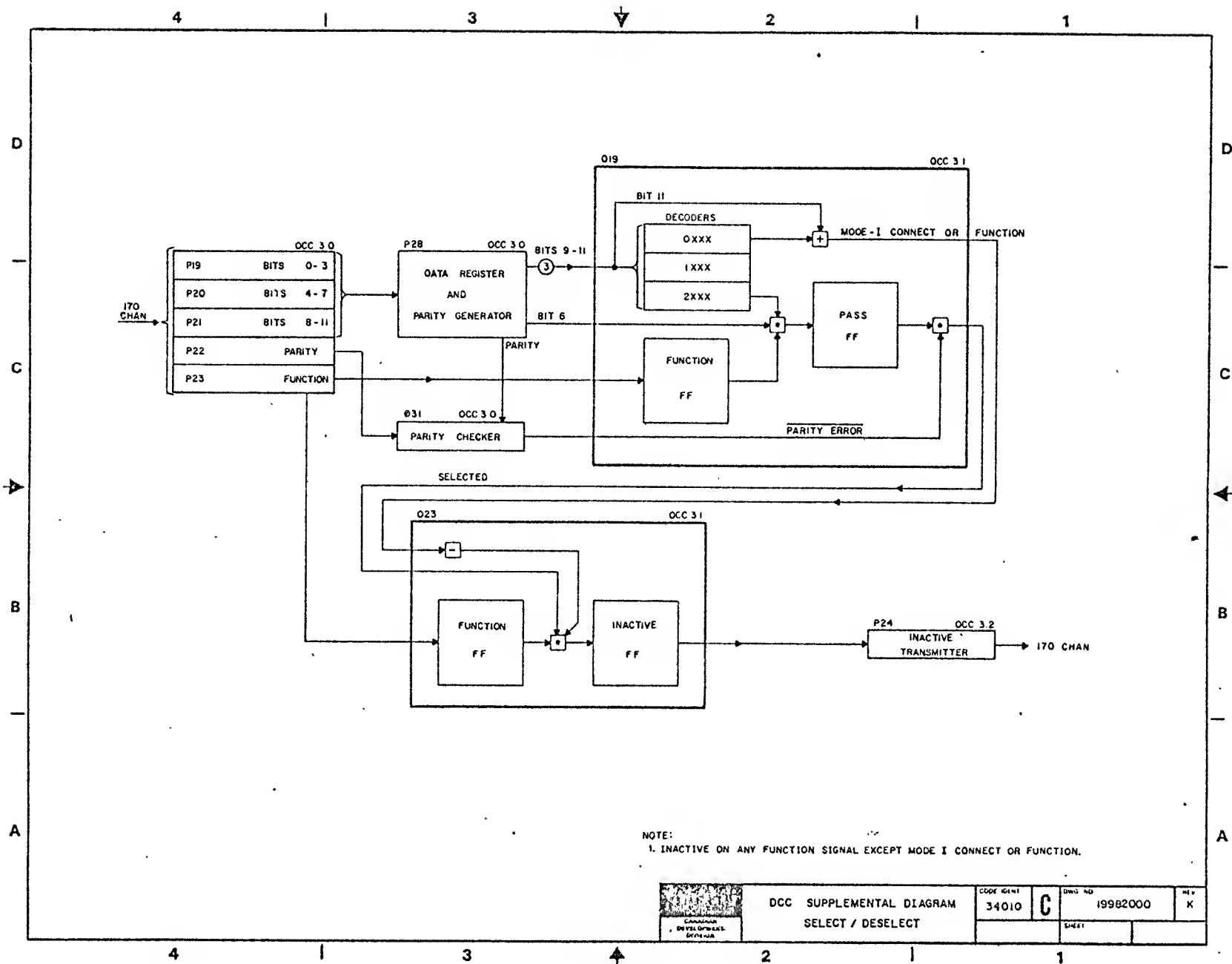
DETAILED PAK DIAGRAM (DCC 3.2)  
INPUT DATA TRANSLATOR

The DS paks (R8 through 14) accept the 3000 series equipment data and control signals, equipment status bits, and the individual interrupts which make up 8 of the 12 converter status bits. It converts these signals to ECL logic levels.

The data, EST and CST bits are passed to the AG paks (R18, R19, R20) where one of the three sets is selected. The selection is dependent upon the signals DV112 and DV113 (12XX and 13XX decoded).

The selected 12 bits are parity checked in the DY pak and simultaneously passed to AP paks for transmission to the channel.

The DY pak (O31) contains the latches for setting converter status bits 02 and 11, and also latches for generating inactive response signals. The converter status codes require bits 02 and 11 to be set upon detection of a parity error on data received from channel.



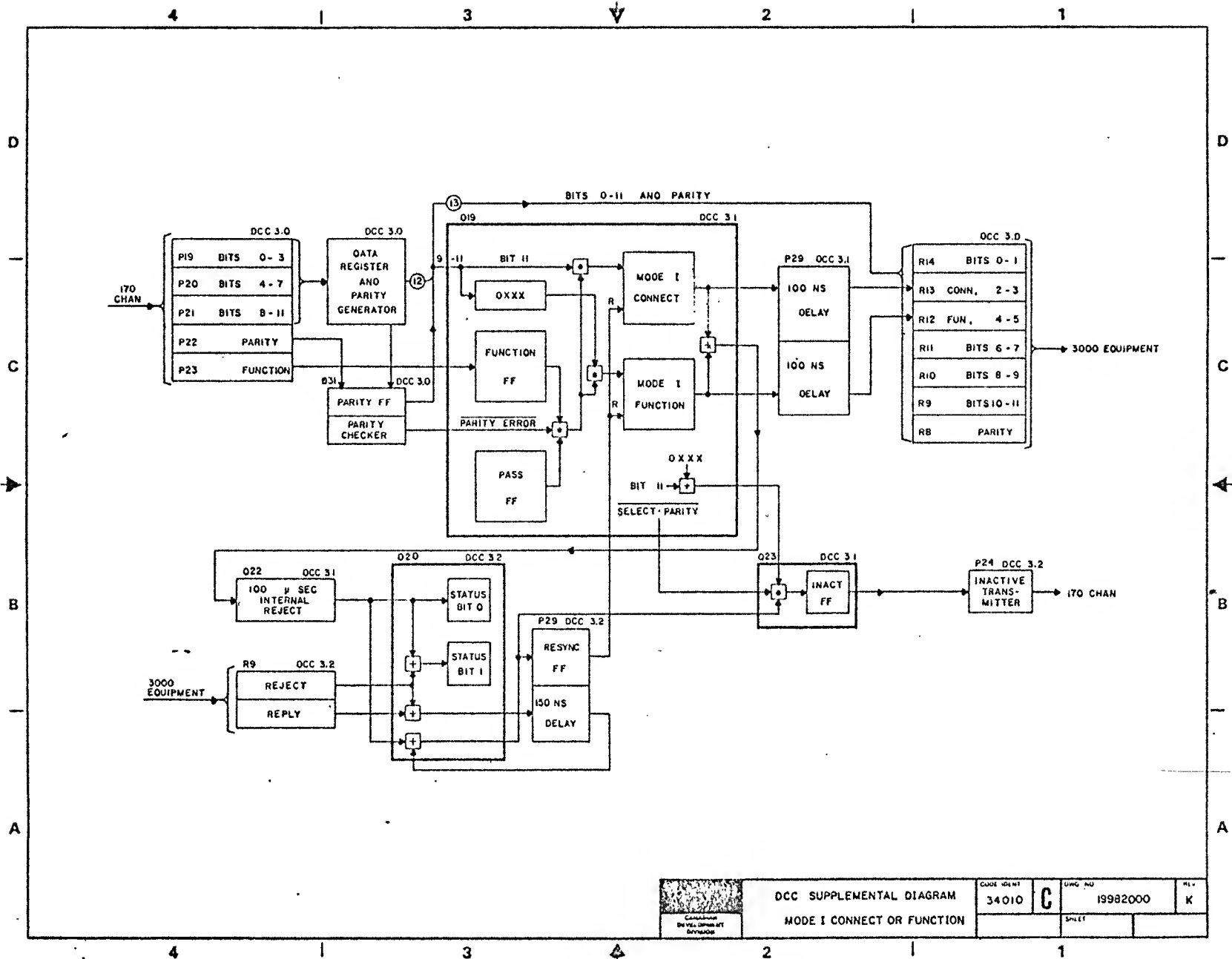
DCC SUPPLEMENTAL DIAGRAM  
SELECT / DESELECT

CODE 34010

C

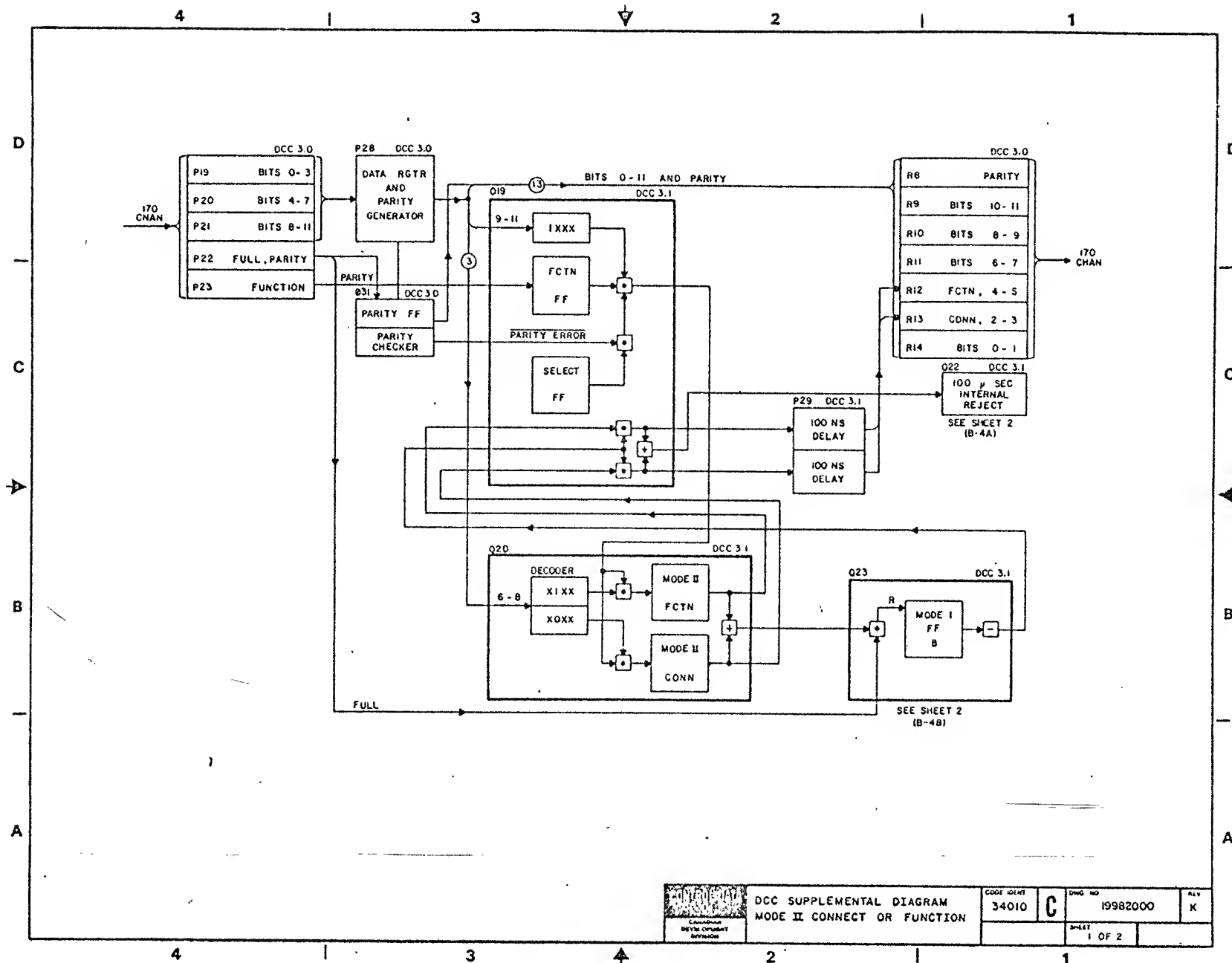
19982000

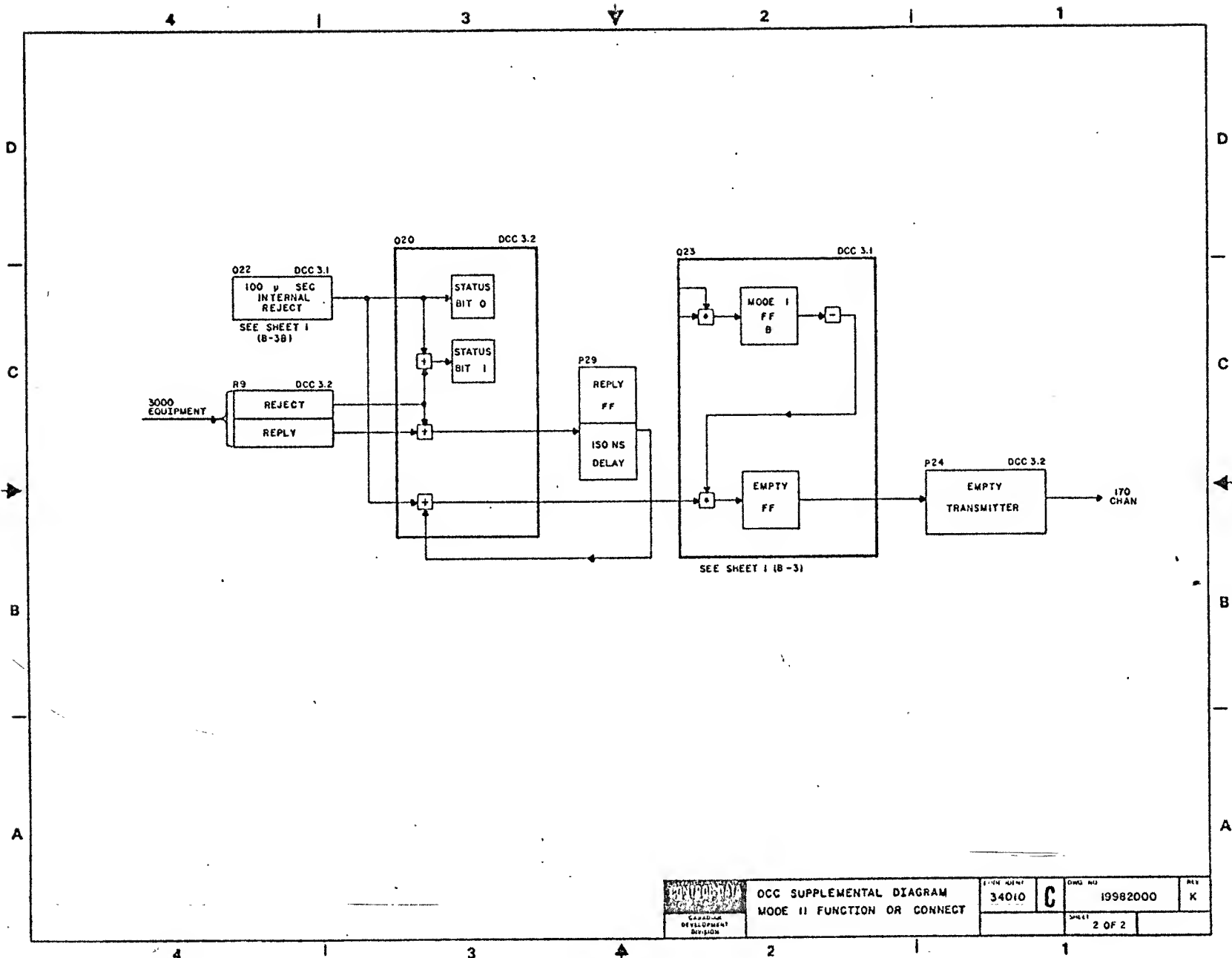
K



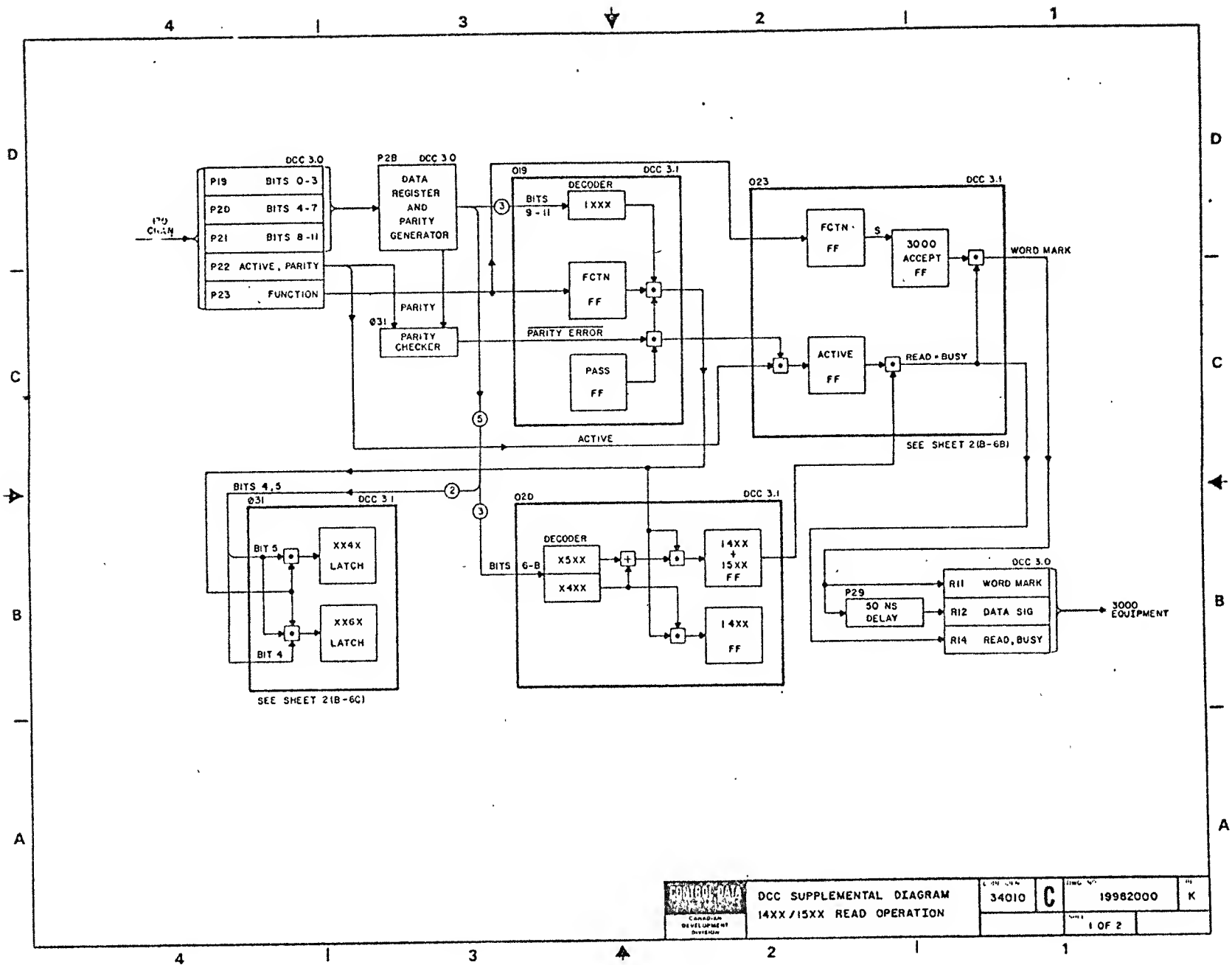
DCC SUPPLEMENTAL DIAGRAM  
MODE I CONNECT OR FUNCTION

CODE IDENT	34010	UMG NO	19982000	REV	K
		SHEET			





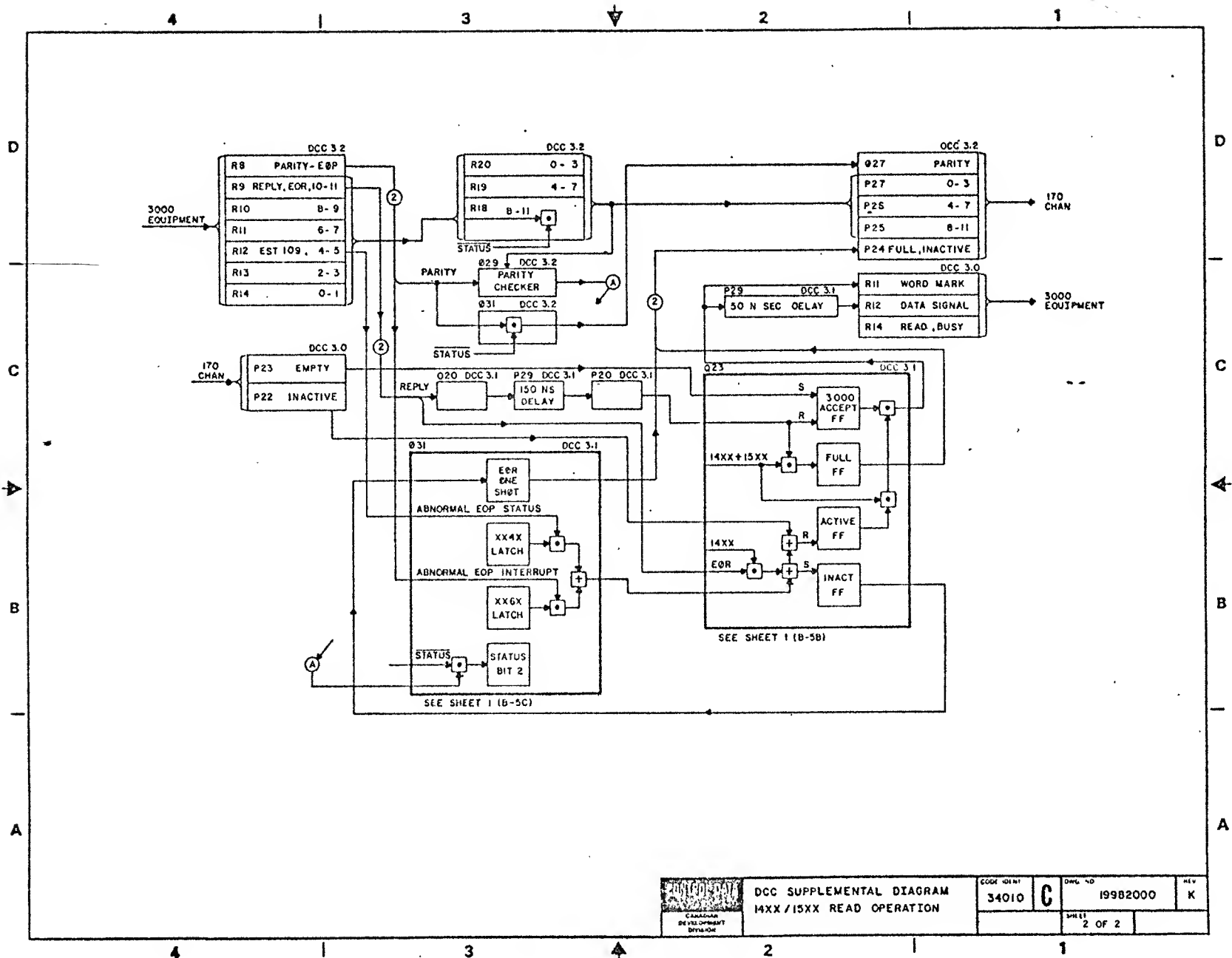
19982000 S

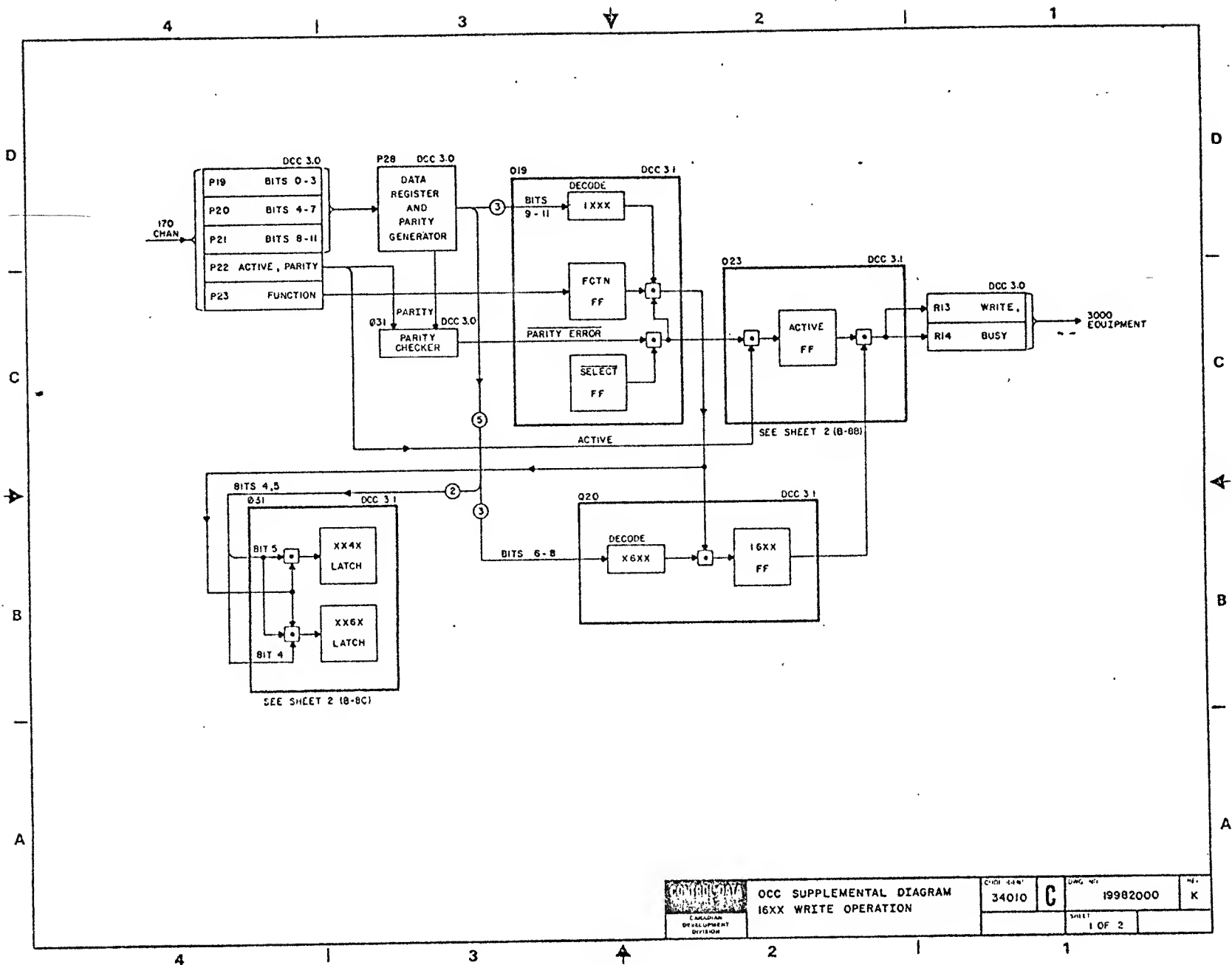


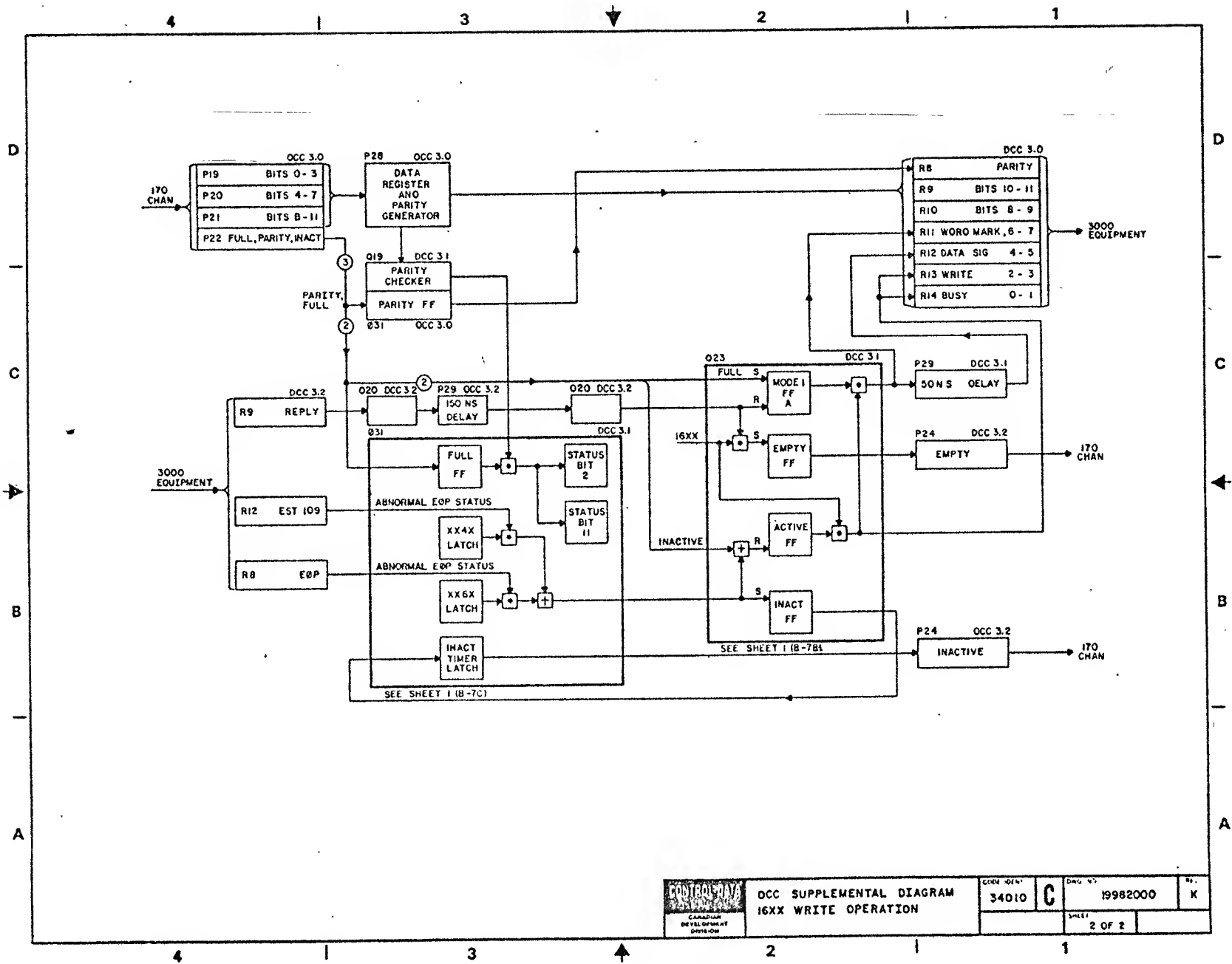
DCC SUPPLEMENTAL DIAGRAM  
14XX/15XX READ OPERATION

34010	C	19982000	K
		1 OF 2	

5-B-5

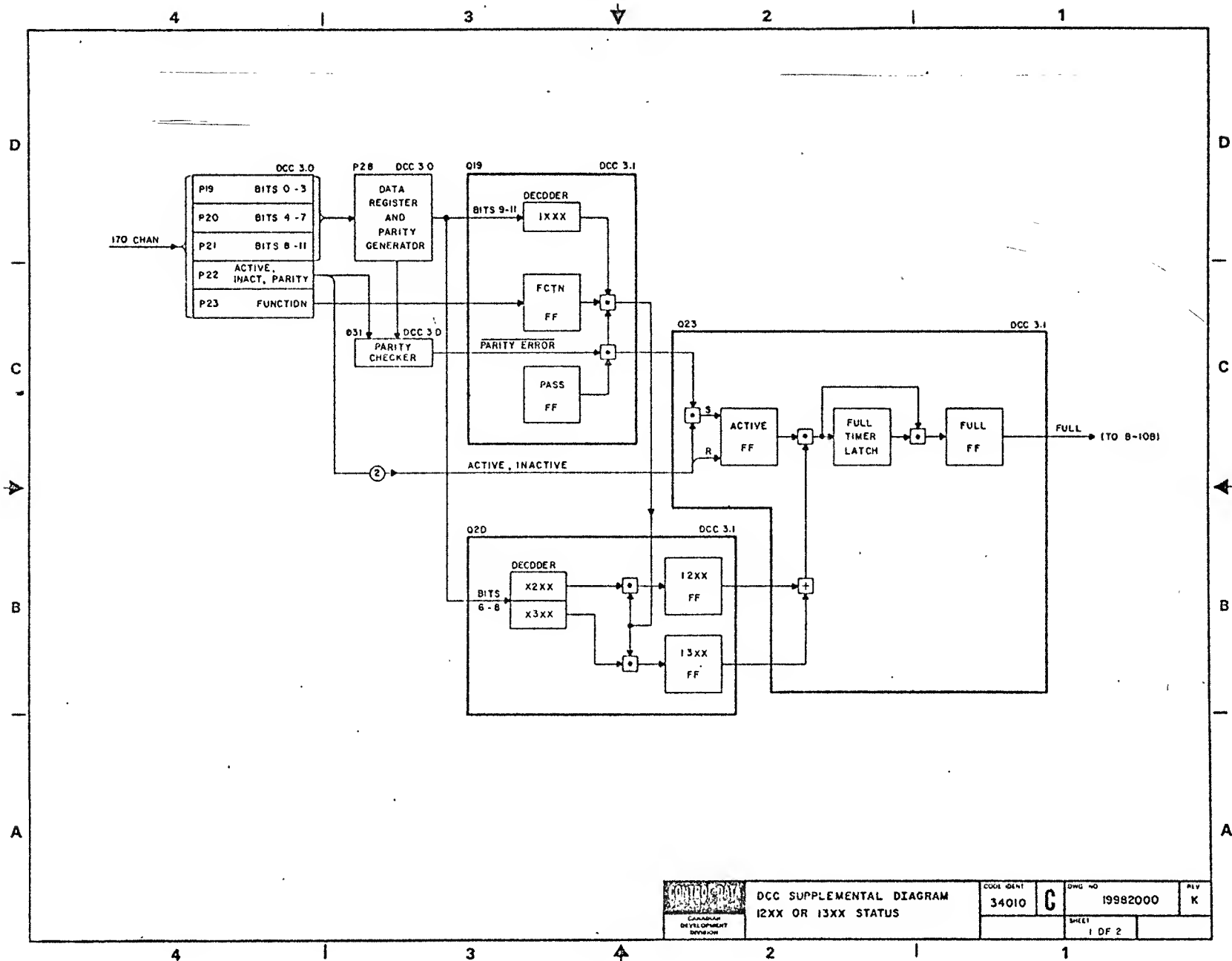






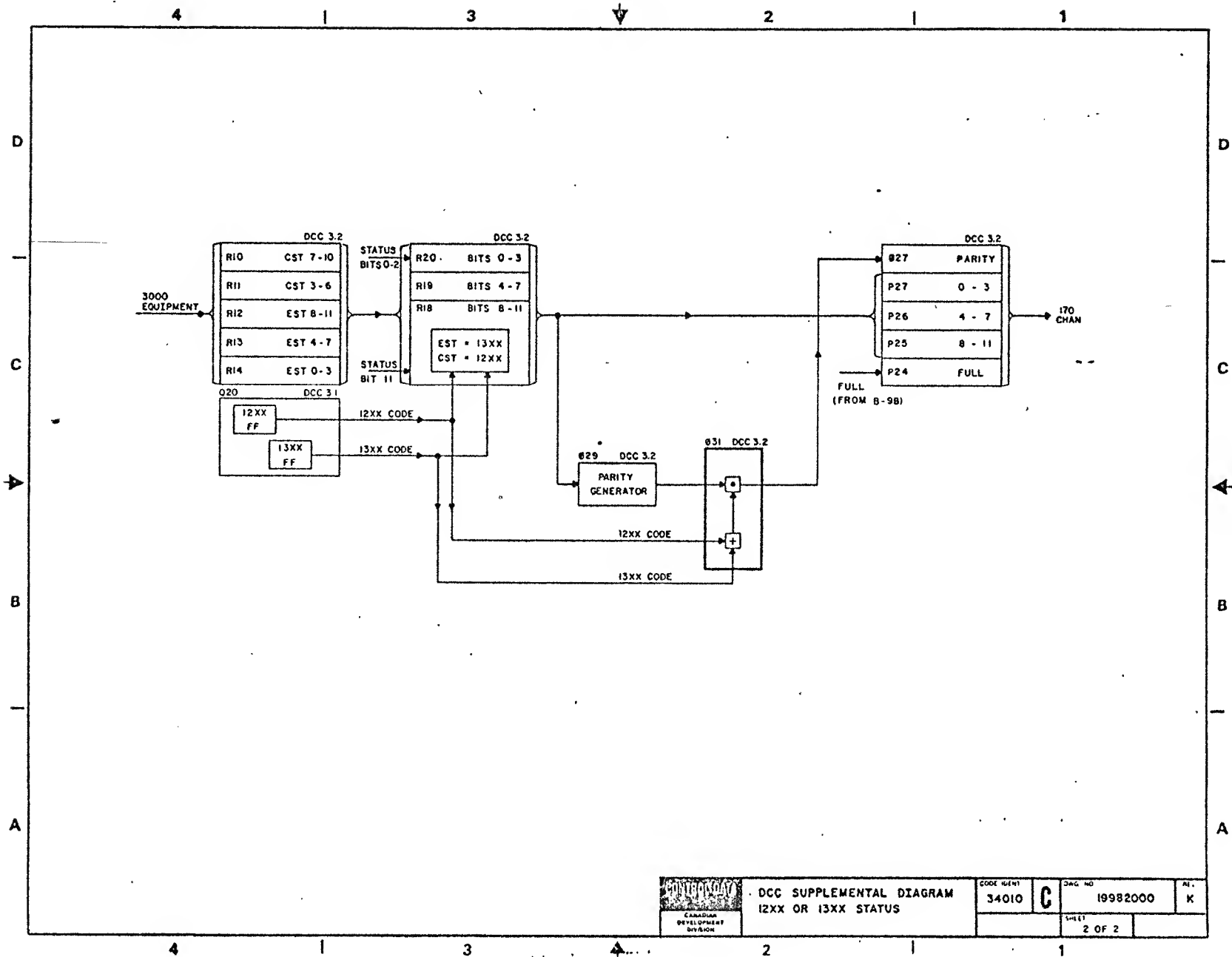
19982000 S

5-B-9



DCC SUPPLEMENTAL DIAGRAM  
12XX OR 13XX STATUS

CODE IDENT 34010	DWG NO 19982000	REV K
SHEET 1 OF 2		



DCC SUPPLEMENTAL DIAGRAM  
12XX OR 13XX STATUS

CODE IDENT  
34010

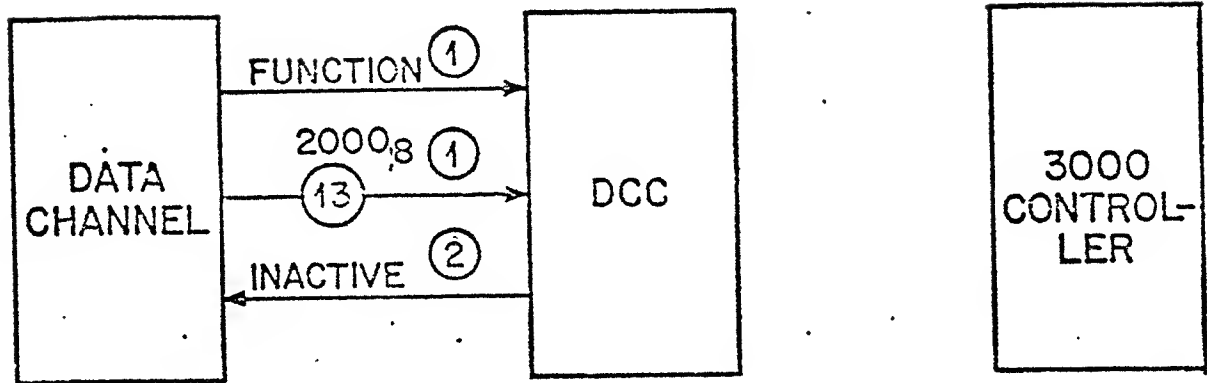
C

DWG NO  
19982000

AL  
K

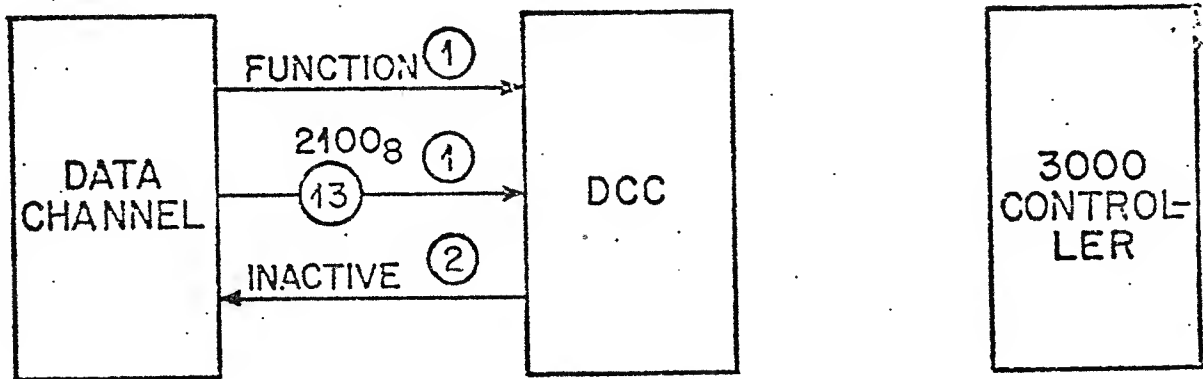
SHEET  
2 OF 2

# DATA CHANNEL CONVERTER SELECT



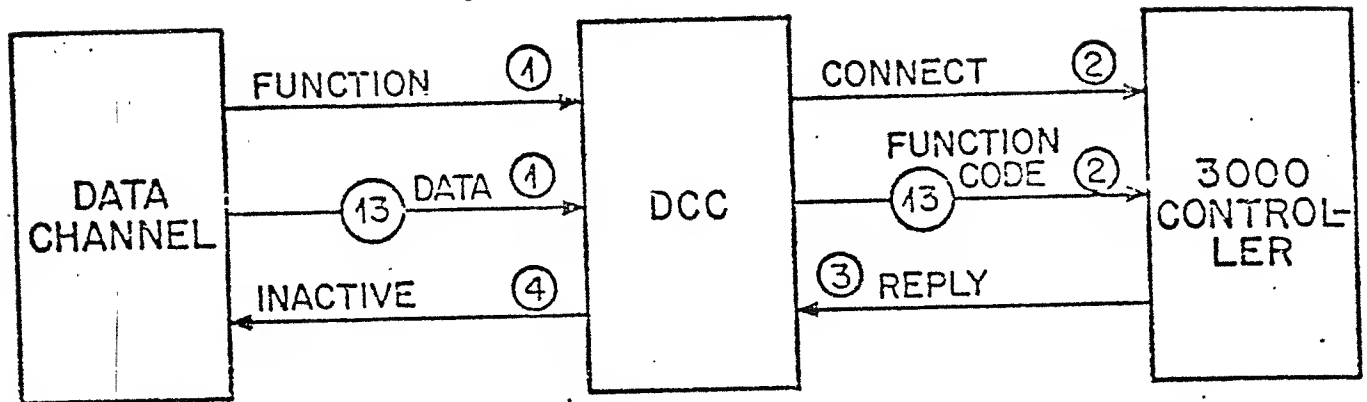
77XX } Function channel XX  
2000 } with 2000<sub>8</sub> code

# DATA CHANNEL CONVERTER DESELECT



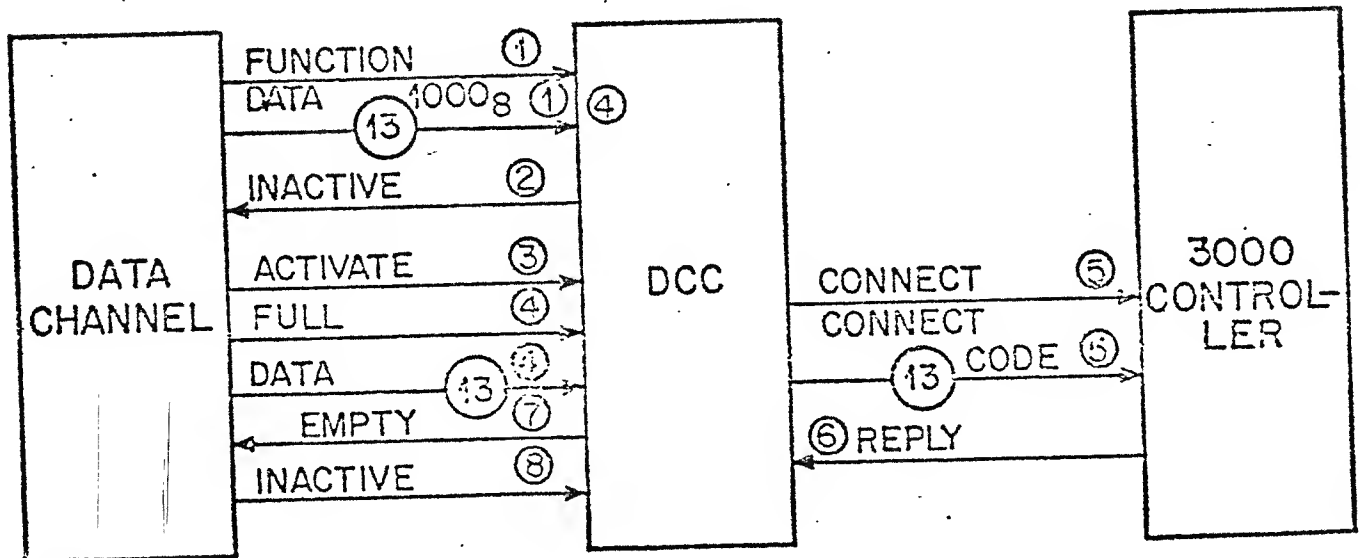
77XX } Function channel XX  
2100 } with 2100<sub>8</sub> code

# CONNECT OPERATION MODE NO. 1



77XX } Function channel XX with connect  
EUUU } code EUUU. E is equipment number  
 and U is unit number

# CONNECT OPERATION MODE NO. 2



77XX } Function code to  
1000 } DDC

2000 } Load A with  
E00U } connect code

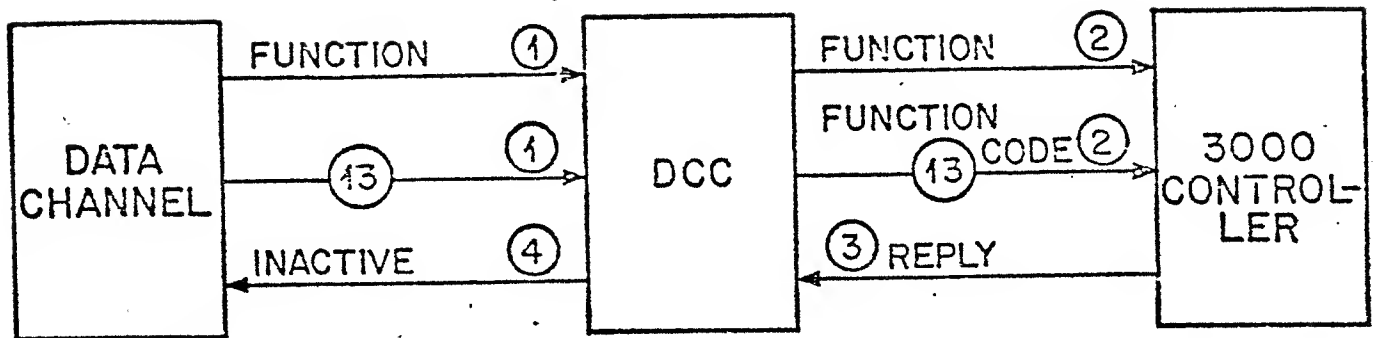
74XX Activate channel XX

72XX Output "1" word from "A" register

M - 66XX } Jump to M if  
M } channel full

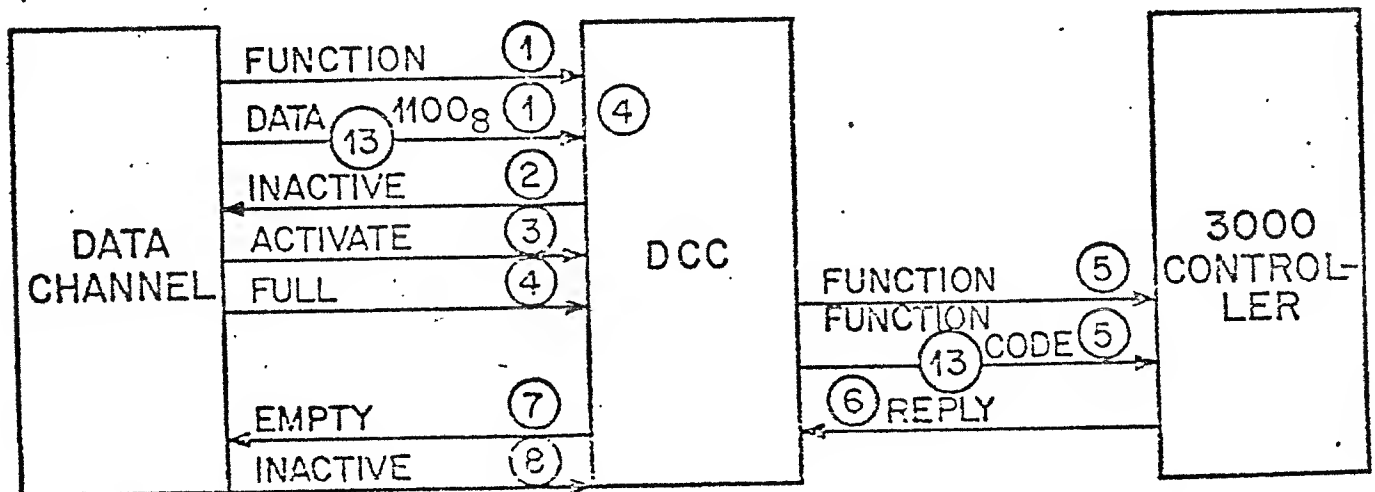
75XX Disconnect channel

# FUNCTION OPERATION MODE NO. 1



77XX } Function channel XX with  
0FFF } function code 0FFF (9 bit code)

# FUNCTION OPERATION MODE NO. 2



77XX } Function code  
1100 } to DCC

2000 } Load A with  
FFFF } 12 bit function code

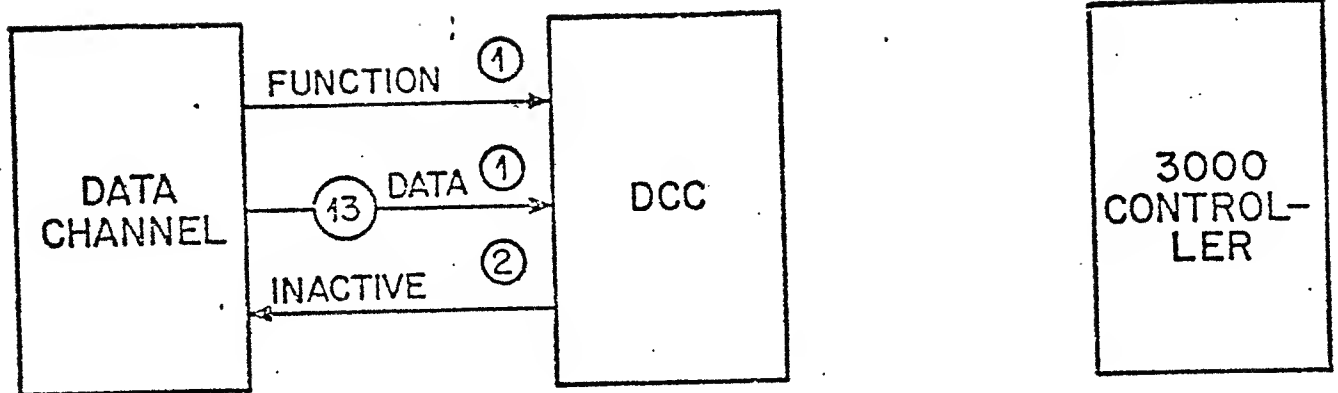
74XX Activate channel XX

72XX Output "1" word from "A" register

M - 66XX } Jump to M if channel XX  
M } full

75XX Disconnect channel XX

## DCC FUNCTION CODES



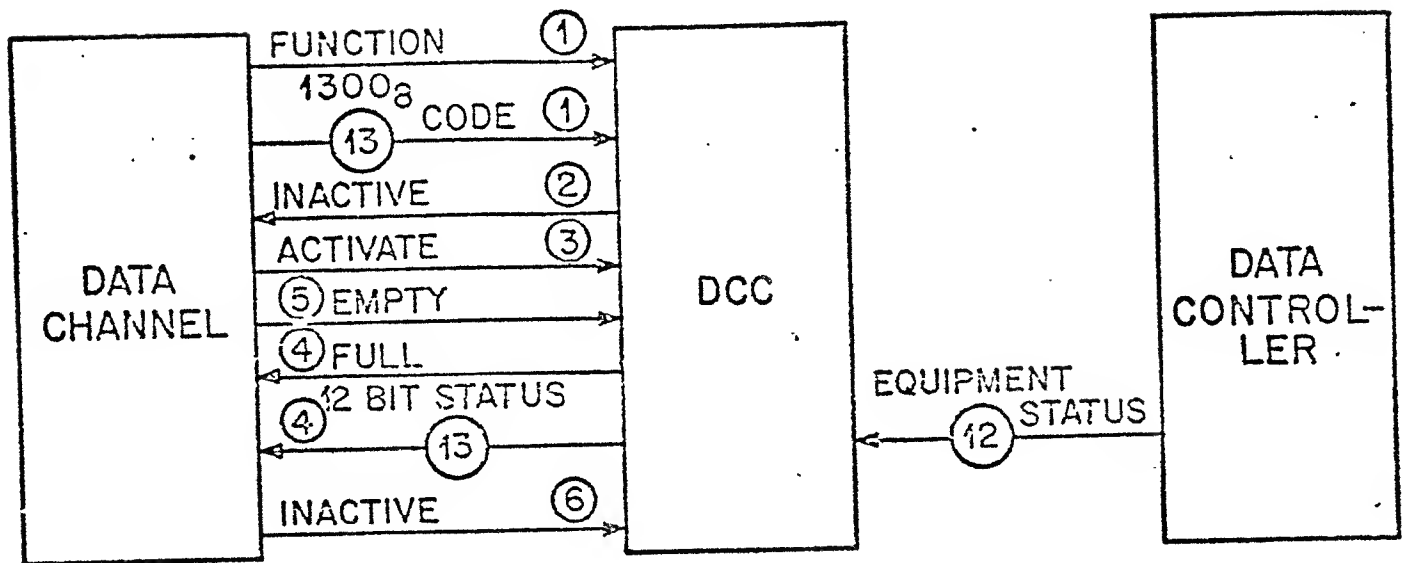
77XX } Function code to  
 YYYY } DCC

### DCC Function Codes

YYYY

- |   |
|---|
| 1600 - Select DCC for an output until inactive                            |
| 1601 - Select DCC for an output until inactive and negate BCD conversion  |
| 1500 - Select DCC for an input until inactive                             |
| 1501 - Select DCC for an input until inactive and negate BCD conversion   |
| 1400 - Select DCC for an input to end of record                           |
| 1401 - Select DCC for an input to end of record and negate BCD conversion |
| 1700 - Send a Master Clear on the channel                                 |

# EQUIPMENT STATUS OPERATIONS



77XX } Function code to  
1300 } DCC

74XX Activate channel XX

70XX Input to "A" from channel XX

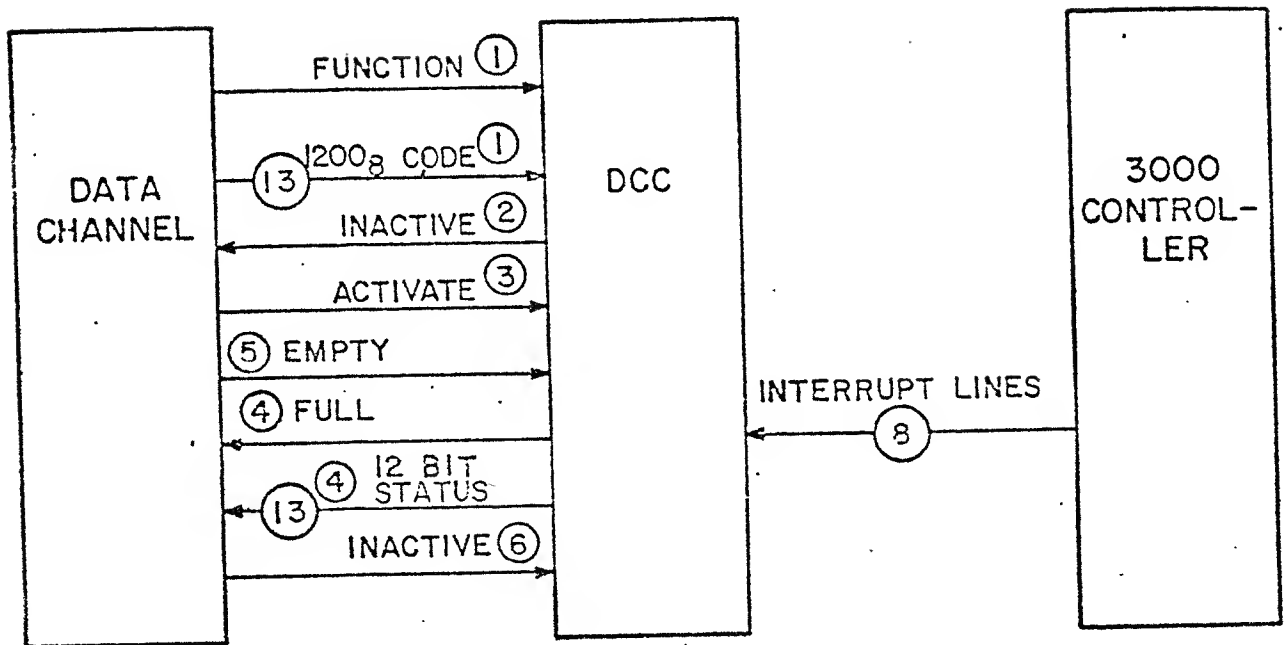
75XX Disconnect channel XX

## Equipment Status Word

11109	876	543	210
XXX	XXX	XXX	XXX

12 bit equipment status word. See 3000 Series  
Computer System Peripheral Equipment Codes  
Publication No. 60113400.

# DCC STATUS OPERATION



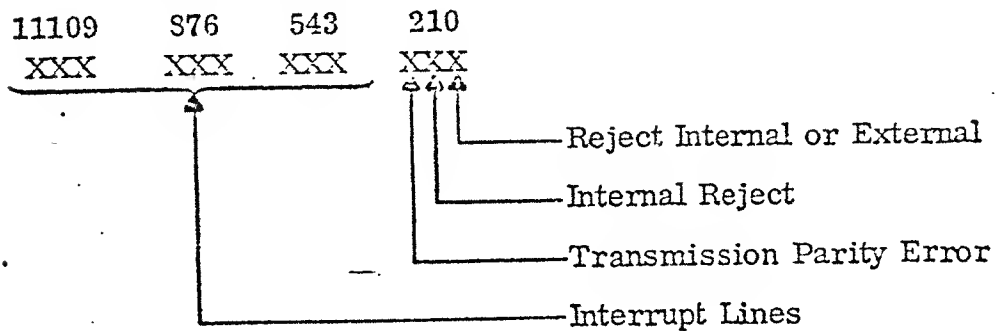
77XX } Function code to  
1200 } DCC

74XX Activate channel XX

70XX Input to "A" from channel XX

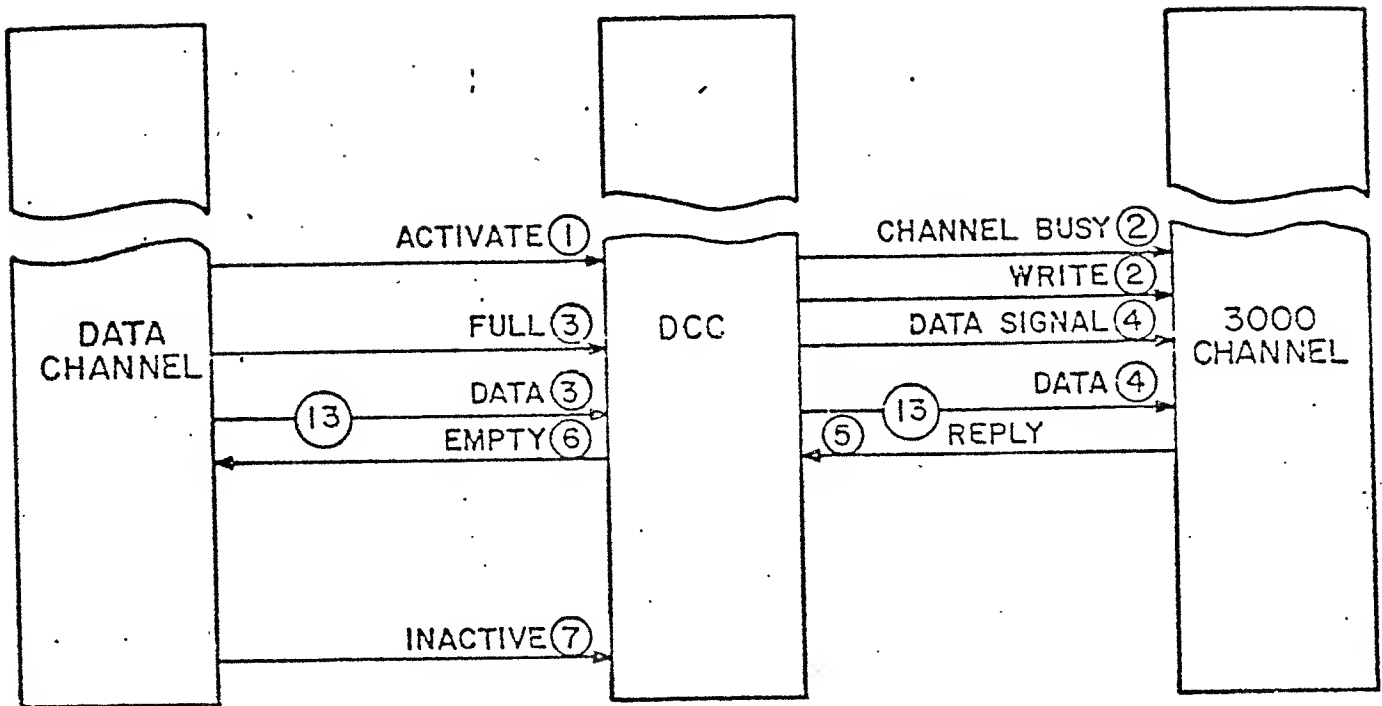
75XX Disconnect channel XX

## DCC Status Word



<u>DCC Status</u>		
<u>Location</u>	<u>Data</u>	
3220	— 0100	
3221	— XXXX	Return Jump Address
3222	— 77XX	Function DCC Status
3223	— 1200	
3224	— 74XX	Activate
3225	— 70XX	Input
3226	— 75XX	Disconnect
3227	— 1207	Logical Product
3230	— 0467	Jump back 10 if "A" equals zero
3231	— 0100	Error Halt
3232	— 0232	

## A BLOCK OUTPUT OPERATION



### Output Operation Program

Check channel inactive **1**

Select DCC

Connect equipment

Check DCC status

Function equipment

Check DCC status

Function DCC for output

Block write operation **2**

See previous operations

#### **1** Inactive Channel Check

65XX	} Jump to M if channel XX inactive
M	

75XX	} Disconnect channel XX
M - ZZZZ	

Steps 3 through 6 will repeat until word count equals zero.

② Block Write Operation

20XX } Load A with  
XXXX } word count

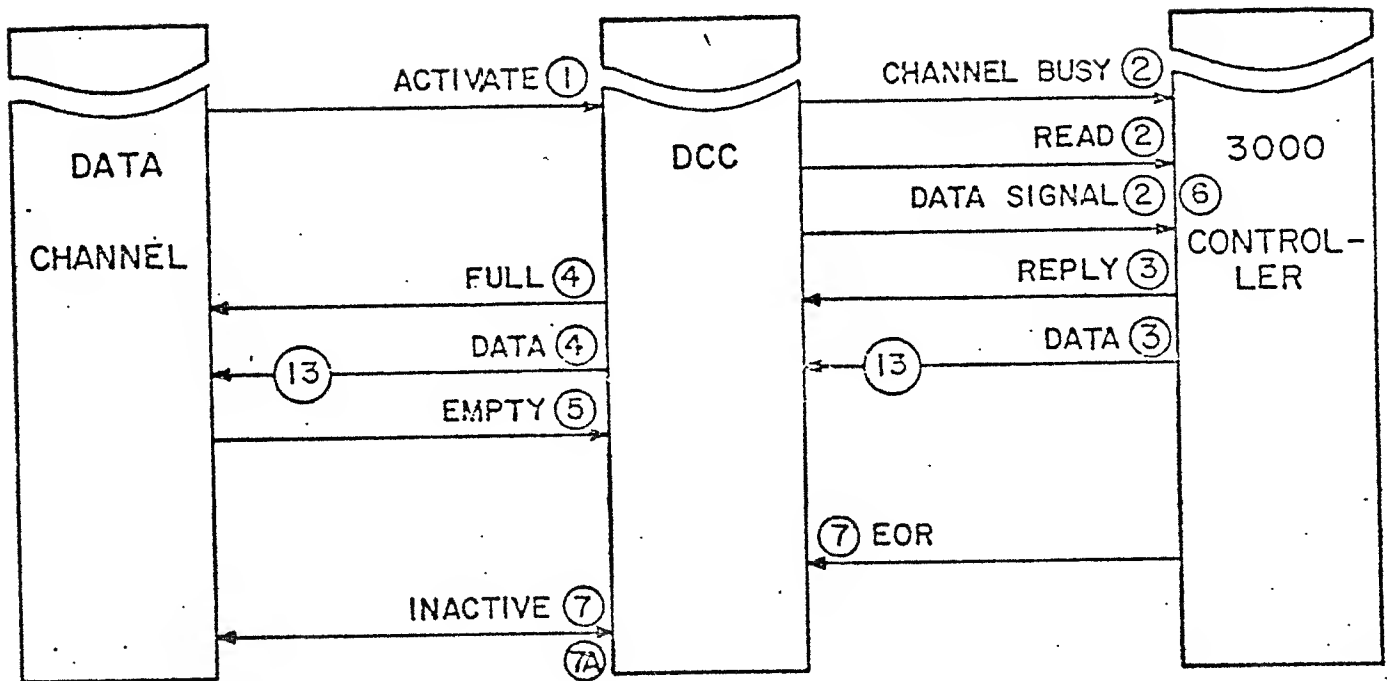
74XX    Activate channel XX

73XX } Block output on channel XX  
YYYY } starting at YYYY

M - 66XX } Jump to M if channel  
M        } XX full

75XX    Disconnect channel XX

## A BLOCK INPUT OPERATION



### Input Operation Program

Check for channel inactive ①

Select DCC

Connect equipment

Check DCC status

Function equipment

Check DCC status

Function DCC for input

Block read operation ②

See previous operations

### ① Inactive Channel Check

65XX	} Jump to M if channel
M	

75XX	} Disconnect channel XX
M - ZZZZ	

Step 3 through 6 will repeat until word count equals zero or EOR

② Block Read Operation

20XX Load A with  
XXXX block count

74XX Activate channel XX

71XX Block input on Channel XX  
YYYY starting at YYYY

\*75XX Disconnect channel XX

\* Not needed if DCC selected to input to end of record.

# I/O DRIVER HANDOUT

## Connect Codes

Equipment	Code
Card Reader	4000
Card Punch	7000
Line Printer	6000

## Function Codes

Equipment	Code	
Card Reader	0001	NEGATE Hollerith to BCD
Card Punch	0001	Select Binary Mode
Line Printer	0004	Page Eject

## Status Checking Routines for Programs

Use the return jump (0200 XXXX) instruction to enter the routine.

### Equipment Status for Ready and Not Busy

Location	Data	
3200	0100	
3201	XXXX	Return Jump Address
3202	77XX	Function DCC for Equipment Status
3203	1300	
3204	74XX	Activate Channel XX
3205	70XX	Input to A from Channel XX
3206	75XX	Disconnect Channel XX
3207	1203	Logical Product Bits 0 and 1
3210	1701	Subject 1 from A
3211	0570	Nonzero Jump Back Seven Places
3212	0100	Long Jump out of Routine
3213	3200	

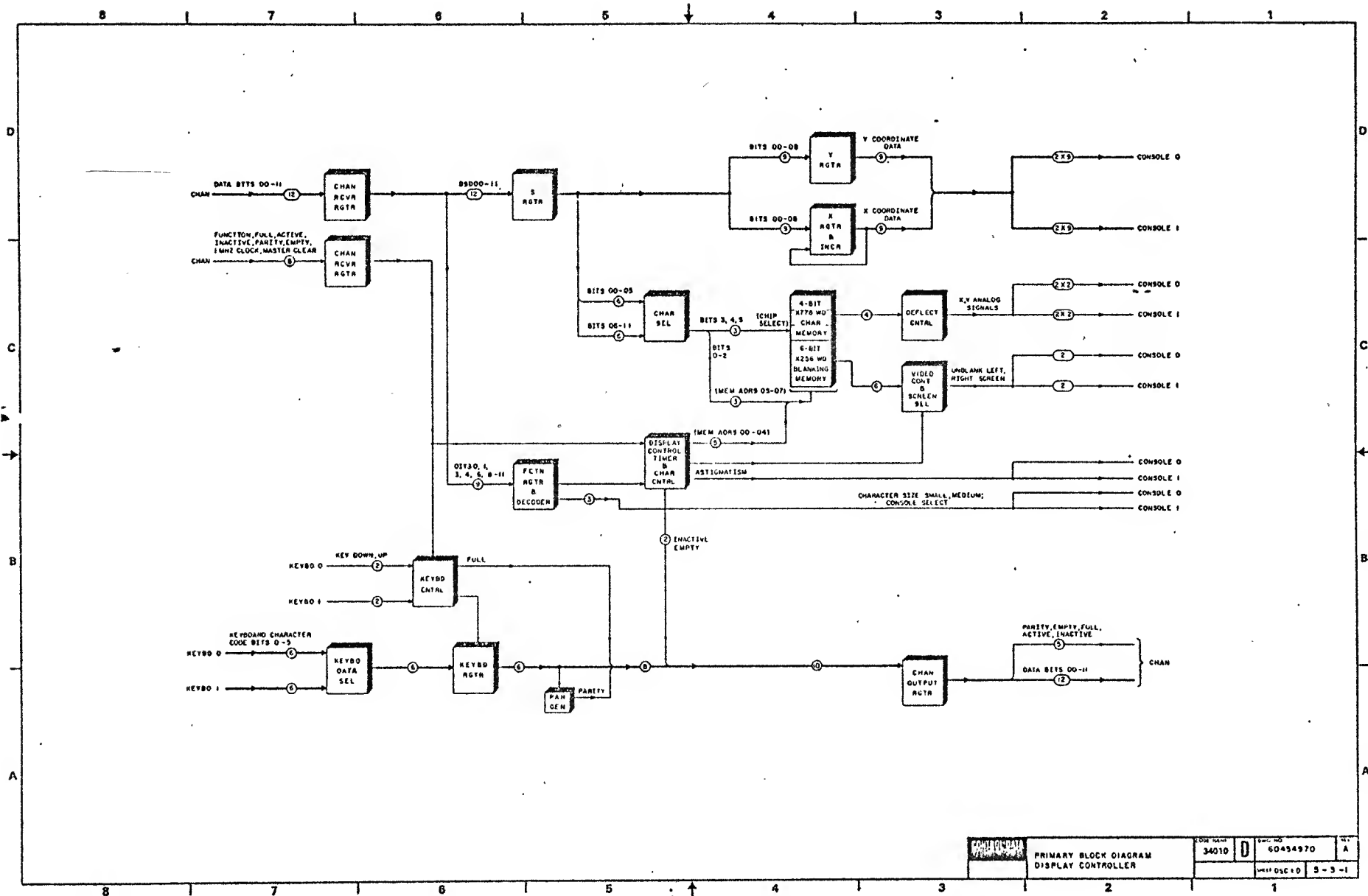


TABLE 4-2-5. CHANNEL INTERFACE

TERM	DESIGNATION
X1000	Channel output bit 0
X1001	Channel output bit 1
X1002	Channel output bit 2
X1003	Channel output bit 3
X1004	Channel output bit 4
X1005	Channel output bit 5
X1006	Channel output bit 6
X1007	Channel output bit 7
X1008	Channel output bit 8
X1009	Channel output bit 9
X1010	Channel output bit 10
X1011	Channel output bit 11
PARY10	Channel output parity bit
FUNY10	Channel output function
FULY10	Channel output full
ACTY10	Channel output active
IACY10	Channel output inactive
EMPY10	Channel output empty
MCLR10	Channel output master clear
1MHZ10	Channel output 1-MHz clock
Z1000	Channel input bit 0
Z1001	Channel input bit 1
Z1002	Channel input bit 2
Z1003	Channel input bit 3
Z1004	Channel input bit 4
Z1005	Channel input bit 5
Z1006	Channel input bit 6
Z1007	Channel input bit 7
Z1008	Channel input bit 8
Z1009	Channel input bit 9
Z1010	Channel input bit 10
Z1011	Channel input bit 11
PAR10	Channel input parity bit
ACT10	Channel input active
FUL10	Channel input full
IAC10	Channel input inactive
EMP10	Channel input empty

PRIMARY BLOCK DIAGRAM (DSC 1.0)

INTRODUCTION

The display controller (DSC) is an integral part of the standard PPS chassis and is equipped to control one or two single-tube display stations.

The DSC generates graphical (dot mode) or alphanumeric (character mode) displays. Alphanumeric characters, which are generated within the DSC, can be presented in one of three sizes. In the standard PPS chassis, the controller is hard-wired to I/O channel 10<sub>8</sub> (except in Model B where pass-on is not provided).

INPUT INTERFACE (refer to table 4-2-5)

Input signals from the PPS to the DSC and vice versa are at standard I/O channel levels.

The clocks used by the DSC are connected directly from internal chassis sources.

OUTPUT INTERFACE (refer to table 4-2-6)

With the exception of the X and Y analog signals, all interfacing with the display console is in positive logic at TTL levels (i.e., 0 volts nominal = logical 0 and +3 volts nominal = logical 1). The X and Y analog signals are differential, with excursions between +0.2V and +2.0V.

OPERATION (refer to DSC 1.0)

A 12-bit function code word from the PPS specifies the operating mode of the DSC. This word is decoded in the function register (refer to table 4-2-7). Segments of the word define the mode as follows:

(X = a don't care bit)

Equipment select	7XXX <sub>8</sub>	- DSC addressed
Screen select	X0XX <sub>8</sub>	- display on left screen
	X1XX <sub>8</sub>	- display on right screen
	X4XX <sub>8</sub>	- display on both screens
Display mode	XX0X <sub>8</sub>	- character
	XX1X <sub>8</sub>	- dot
	XX2X <sub>8</sub>	- keyboard input request
Character size	XXX0 <sub>8</sub>	- small
	XXX1 <sub>8</sub>	- medium
	XXX2 <sub>8</sub>	- large

Note: When set in an operating mode, the DSC will treat all subsequent data in a similar manner. Change of mode requires the receipt of another function code.

TABLE 4-2-6. CONSOLE INTERFACE

TERM	DESIGNATION
XDEF00	X coordinate bit 0
XDEF01	X coordinate bit 1
XDEF02	X coordinate bit 2
XDEF03	X coordinate bit 3
XDEF04	X coordinate bit 4
XDEF05	X coordinate bit 5
XDEF06	X coordinate bit 8
XDEF07	X coordinate bit 7
XDEF08	X coordinate bit 8
YDEF00	Y coordinate bit 0
YDEF01	Y coordinate bit 1
YDEF02	Y coordinate bit 2
YDEF03	Y coordinate bit 3
YDEF04	Y coordinate bit 4
YDEF05	Y coordinate bit 5
YDEF06	Y coordinate bit 6
YDEF07	Y coordinate bit 7
YDEF08	Y coordinate bit 8
SCSEL	Screen select
UBLAKL	Unblank left screen
UBLAKR	Unblank right screen
SMALL	Character size small
MEDIUM	Character size medium
ASTM	Astigmatism
DCON00	Keyboard output bit 0
DCON01	Keyboard output bit 1
DCON02	Keyboard output bit 2
DCON03	Keyboard output bit 3
DCON04	Keyboard output bit 4
DCON05	Keyboard output bit 5
DCONKD	Keyboard key down
DCONKU	Keyboard key up
XANLG1	X plane analog signal 1
XANLG2	X plane analog signal 2
YANLG1	Y plane analog signal 1
YANLG2	Y plane analog signal 2

## OPERATION IN CHARACTER MODE

1. If the code is in the form  $7XXX_8$ , the 9 lower-order bits specify the Y coordinate for beam positioning and will be stored in the Y coordinate register.
2. If the code is in the form  $6XXX_8$ , the 9 lower-order bits specify the X coordinate for beam positioning and will be stored in the X coordinate register.
3. Any data word not in the forms specified above will be interpreted as two 6-bit character codes. The first character code occupies bit positions 11 through 6 and the second character code bit positions 5 through 0. The data word is passed to the character selector and chip decoder where the first character code is selected by command from the timing counter and character control. The character code selects the start point of a series of instructions stored in the programmable read-only memory (PROM). These instructions, addressed incrementally by action of the timing counter and character control move the CRT beam, from its start point as defined by the contents of the X and Y coordinate registers, in such a way as to paint a character on the screen. The PROM produces X and Y deflection and CRT beam on/off instructions. On completion of the character paint the X coordinate register is incremented to position the beam correctly for the next character. The whole process is repeated for the second half of the data word.

TABLE 4-2-7. FUNCTION CODES

Equipment Select	7---	Select Console Display
Screen	-0--	Left screen
	-1--	Right screen
	4	Both screens
Mode	--0-	Character mode
	--1-	Dot mode
	--2-	Keyboard input request
Character size	---0	64 characters/line (small)
	---1	32 characters/line (medium)
	---2	16 characters/line (large)
<p>NOTE</p> <p>For dot mode and keyboard input request, bits 0-2 are ignored and generally are 000.</p>		

## COORDINATES

The active area of the CRT screen is divided into 512 equispaced horizontal lines, each with 512 addressable dot positions. Each dot position is uniquely defined by a 9-bit X coordinate and a 9-bit Y coordinate;  $X = 000_8; Y = 000_8$  being at the bottom left-hand corner of the display and  $X = 777_8; Y = 777_8$  at the top right-hand corner. A small character is defined within an  $8 \times 8$  coordinate matrix, medium  $16 \times 16$  and large  $32 \times 32$ . This means that a maximum of 16 rows of 16 large characters, 32 rows of 32 medium characters or 64 rows of 64 small characters can be displayed on the CRT screen.

TABLE 4-2-8. CONSOLE DISPLAY AND KEYBOARD CHARACTER CODES

Character	Code	Character	Code	Character	Code
(space)	00	P	20	5	40
A	01	Q	21	6	41
B	02	R	22	7	42
C	03	S	23	8	43
D	04	T	24	9	44
E	05	U	25	+	45
F	06	V	26	-	46
G	07	W	27	*	47
H	10	X	30	/	50
I	11	Y	31	(	51
J	12	Z	32	)	52
K	13	0	33	blank key	53
L	14	1	34	=	54
M	15	2	35	blank key	55
N	16	3	36	,	56
O	17	4	37	.	57
<p>*Keyboard codes are identical to character codes with the following additions and one exception (space):</p> <p>No data.....00      Backspace.....61</p> <p>Carriage return....60      Space.....62</p>					

## CHARACTER GENERATION

Character generation consists of horizontal and vertical beam movements and beam blanking and unblanking. These movements begin at the point on the CRT screen specified by the contents of the X and Y registers. Instructions that specify beam movements for specific characters are stored in a number of preprogrammed read-only memories; the start address for a series being specified by the character code. This address is incremented by a counter, producing the required series. On completion of the character paint, the X coordinate address is incremented by  $8(2^3)$  for small characters,  $16(2^4)$  for medium size characters or  $32(2^5)$  for large characters. This correctly sets the start position for the next paint. (Refer to table 4-2-8.)

## OPERATION IN DOT MODE

In dot mode, the controller responds to  $6XXX_8$  and  $7XXX_8$  codes only. On receipt of a  $7XXX_8$  code, a dot is painted on the screen at the reference point defined by it and the contents of the  $6XXX_8$  (X coordinate) register.

## KEYBOARD INPUT MODE

In this mode a previously loaded 6-bit character code is transmitted from a holding register to the PPS, and the holding register is cleared.

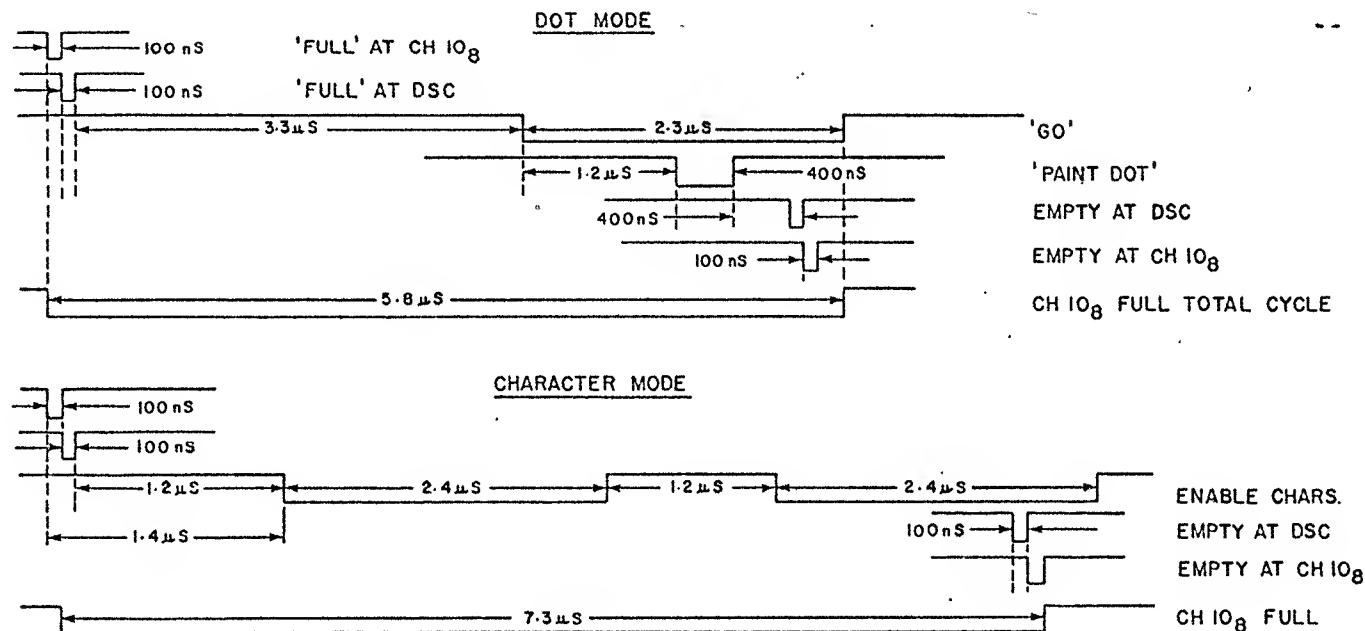


Figure 4-2-15. Timing Cycles for Dot and Character Modes

## DETAILED PAK DIAGRAM (DSC 3.0)

### INPUT, X AND Y REGISTERS

#### INPUT

The Ap registers hold and synchronize the input word and control signals. The 9 relevant bits of the input word are applied to the function register (DD pak). When the function (DCHFUN) line is active, the input code is examined to see if the display controller is being addressed (i.e., the code is 7XXX). If this condition is satisfied, the select equipment signal (DDINAC) is raised with the result that six control lines determining the DSC's mode of operation are enabled.

Bits 00 and 01 govern character size. Bit 04, if active, sets the keyboard select latch. If the keyboard is not selected, the bit 03 determines whether the DSC operates in dot or character mode. The screen select translator (DSC 3.1) interprets bits 6 and 8 to select either the left or right screens, or both, for display. All six control lines are latched and will not change until another function code is received.

After a function is processed, the DCHFUL signal enables the set X and Y coordinate FFs. Completion of such processing is flagged by the setting of equipment select FFs and the absence of DCHFUN signal. A  $6XXX_8$  or  $7XXX_8$  code then sets the X and Y latches, respectively, thereby loading the XXX portion of the input code into the appropriate register. When either latch is set, the DDXYON flag is raised.

With the X or Y coordinate FFs set, the DDSEN signal enables the S register to load the complete 12-bit code word. Only 9 least significant bits are used to specify X and Y coordinates.

#### X AND Y REGISTERS

The display area of the screen is divided into 512 horizontal lines, each with 512 dot locations on it. Each of these positions has been allocated an X coordinate (horizontal) and a Y coordinate (vertical). The address  $X = 000_8; Y = 000_8$  is located at the bottom left-hand corner of the screen, and  $X = 777_8; Y = 777_8$  at the top right-hand corner. These bit addresses are stored in the coordinate registers.

In the dot mode, the entry of a Y coordinate forces the painting of a dot at the location specified by it and the contents of the X register.

In the character mode, the contents of the X and Y registers specify the start point for a character paint. On completion of the paint, an increment circuit increases the address in the X coordinate register by an amount dependent upon the character size selected. Small =  $8_{10}$ ; medium =  $16_{10}$ ; large =  $32_{10}$ . This positions the beam for the next character paint.

DETAILED PAK DIAGRAM (DSC 3.1)  
CHARACTER READ, DEFLECTION AND SCREEN CONTROL

CHARACTER GENERATION AND SELECTION

The instructions for painting each character are stored in programmable read-only memories (PROMs). Six of these (three on each DI1 pak) contain the beam deflection instructions and two (one on each DI1 pak) contain the beam blank/unblank instructions. Each character is painted within an 8 x 8 matrix and the distance between each point on the matrix is governed by the character size selection.

Each PROM contains the paint instructions for eight characters. There are six character PROMs in the system and one of these will be addressed by the 3 most significant bits of the character code. The 3 least significant bits of the code, address the start of one of eight series of paint instructions stored in that PROM. On receipt of a GO signal (DRSTR), the address is incremented by the output of a counter and the PROM produces the series of instructions required to paint the selected character. (See table 4-2-9.)

There are four parallel outputs from each PROM. Those from the character PROMs are designated V1, V2, H1 and H2. An output from V1 alone is an instruction to increment the matrix address one point in the vertical direction; V2 alone increments the vertical address two points. When V1 and V2 occur at the same time, it is interpreted as an instruction to

change the direction of following vertical plane instructions (i.e., from up to down or vice versa). The H outputs are interpreted in a similar manner but apply to the horizontal plane. (See figure 4-2-16.)

The blank/unblank PROMs are addressed by the 3 least significant bits of the character code and produce six streams of blank/unblank pulses.

BLANKING PULSE SELECTION

The correct stream of blank/unblank pulses is selected in the DG pak by the 3 most significant bits of the character code. A further selection of blanking pulse stream or single dot pulse (dot mode) is made by DEDOT. The final selector on this pak chooses left, right, or both screens.

DA CONVERSION OF BEAM DEFLECTION CODE

The DC paks decode the vertical and horizontal instructions (DI1U and DHH signals, respectively) and sum them with the results of previous instructions in the same character point. This decode represents beam position relative to the character's starting point. An increment by one instruction will raise one of the output lines while increment by two raises two lines. Similarly, decrement instructions deactivate one or two lines. There are six output lines each for horizontal and vertical deflection; the total active in the six represents beam position relative to a character's starting point.

Each character sequence begins with a DRSTR signal that clears the output lines to all zeros and sets the increment/decrement condition to increment. Thereafter the coincidence of the by one and by two instructions [(DIH1, DHV2) or (DHH1, DHH2)] is the means of toggling the increment/decrement condition on the vertical and horizontal axis, respectively.

For example, the series of instructions DHV2, DHV2, DIH1, (DIH1, DHV2), DHV2 causes the beam to move successively up two, up two, up one, change direction and down two, thereby leaving the beam placed three positions vertically from the starting point. Three of the DCV 0-5 signals are activated in this situation.

The six output lines are applied to summing networks in the DQ paks where analog driver signals are produced for the CRT.

TABLE 4-2-9. PROGRAMMABLE READ-ONLY MEMORIES

CDC PART NO.	PAK TYPE & LOCATION	ADDRESSED BY	DPAD05 06 07	0	1	0	1	0	1	0	1	0	1
				0	0	1	1	0	0	1	1	0	1
				0	0	0	0	1	1	1	1	1	1
19182200	DJ(R41) - B6	DFCE00		A	B	C	D	E	F	G			
01	DJ(R41) - A6	01		H	I	J	K	L	M	N	O		
02	DJ(R41) - A1	02		P	Q	R	S	T	U	V	W		
03	DH(R42) - B6	03		X	Y	Z	0	1	2	3	4		
04	DH(R42) - A6	04		5	6	7	8	9	+	-	*		
05	DH(R42) - A1	05		/	[	]		=		.	.		
06	DJ(R41) - B1	UNBLANK PULSES FOR	{	19182200 - 03									
07	DH(R42) - B1			19182204 - 05									

C M

## INTRODUCTION

### SYSTEM BLOCK DIAGRAM (figures 5-1-1, 5-1-2)

Central memory for Models 171, 172, 173, 174 consists of one central storage unit (CSU-0); a second unit (CSU-1) is available as an option for the Models 173 and 174. Basic size of CSU-0 is 65K words for the Model 171, basic size of CSU-0 is 32K words for the Model 172, and basic size of CSU-0 and CSU-1 is 65K words for the Models 173 and 174.

Central memory for Model 175 consists of two central storage units (CSU-0 and CSU-1), each having a basic size of 32K words.

Both CSU-0 and CSU-1 are expandable to 131K words.

All communications between central memory and the rest of the system pass through the central memory control (CMC).

Signal lines to each CSU from the CMC include the following:

- 68 Data bits (including 8 SECDED bits)
- 14 Address bits plus 1 parity bit
- 8 'GO' signals
- 8 'GO REFRESH' signals
- 1 'M173' Delayed Write Control signal
- 1 Master Clear signal
- 1 Maintenance Disable signal
- 2 Clock signals (T1 and T3).

Signals from each CSU to the CMC are:

- 68 Data bits (including 8 SECDED bits)
- 1 Address Parity Error signal
- 1 Memory Disabled signal.

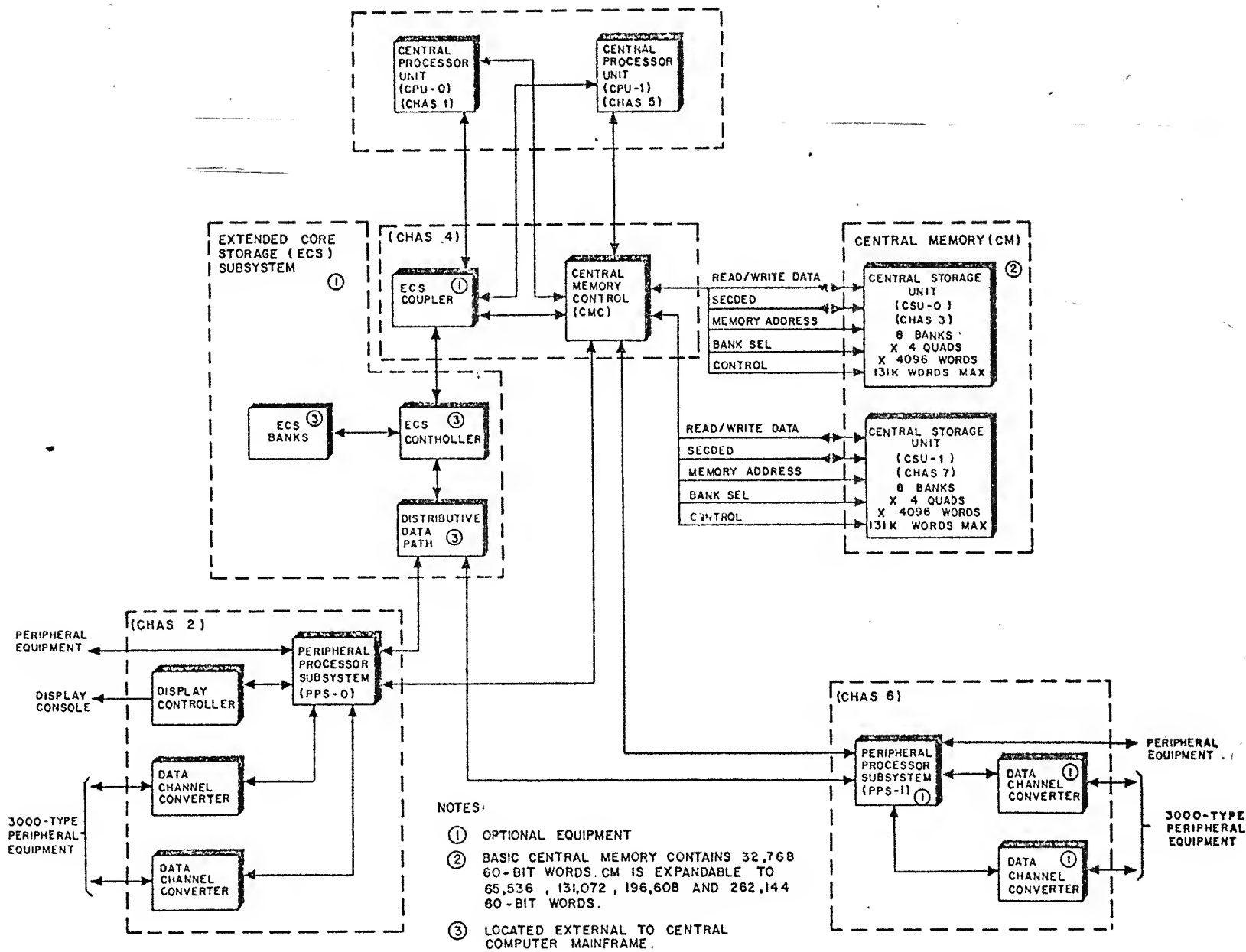


Figure 5-1-1. System Block Diagram: AA107-A, AA131-B (Models 171 through 174)

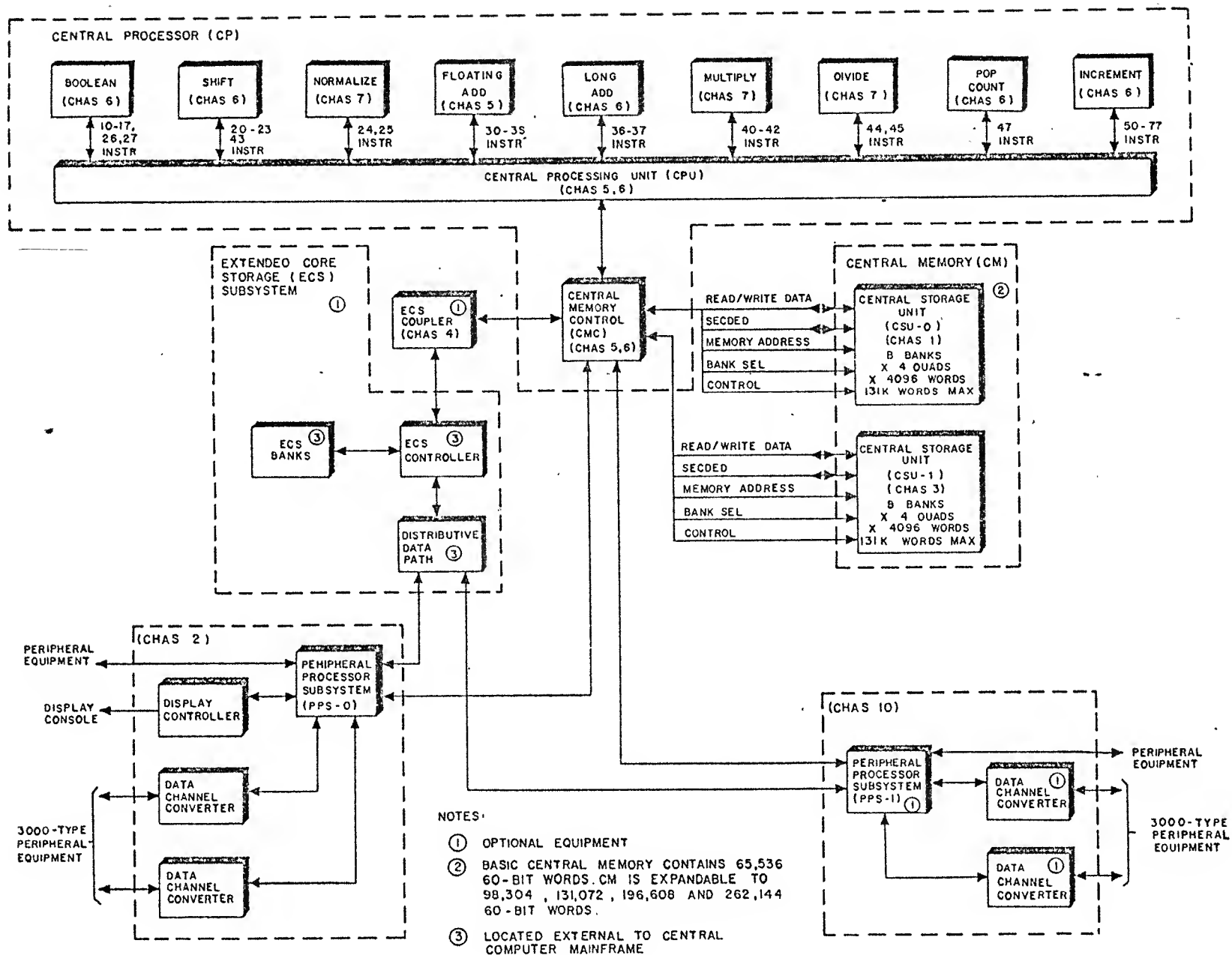


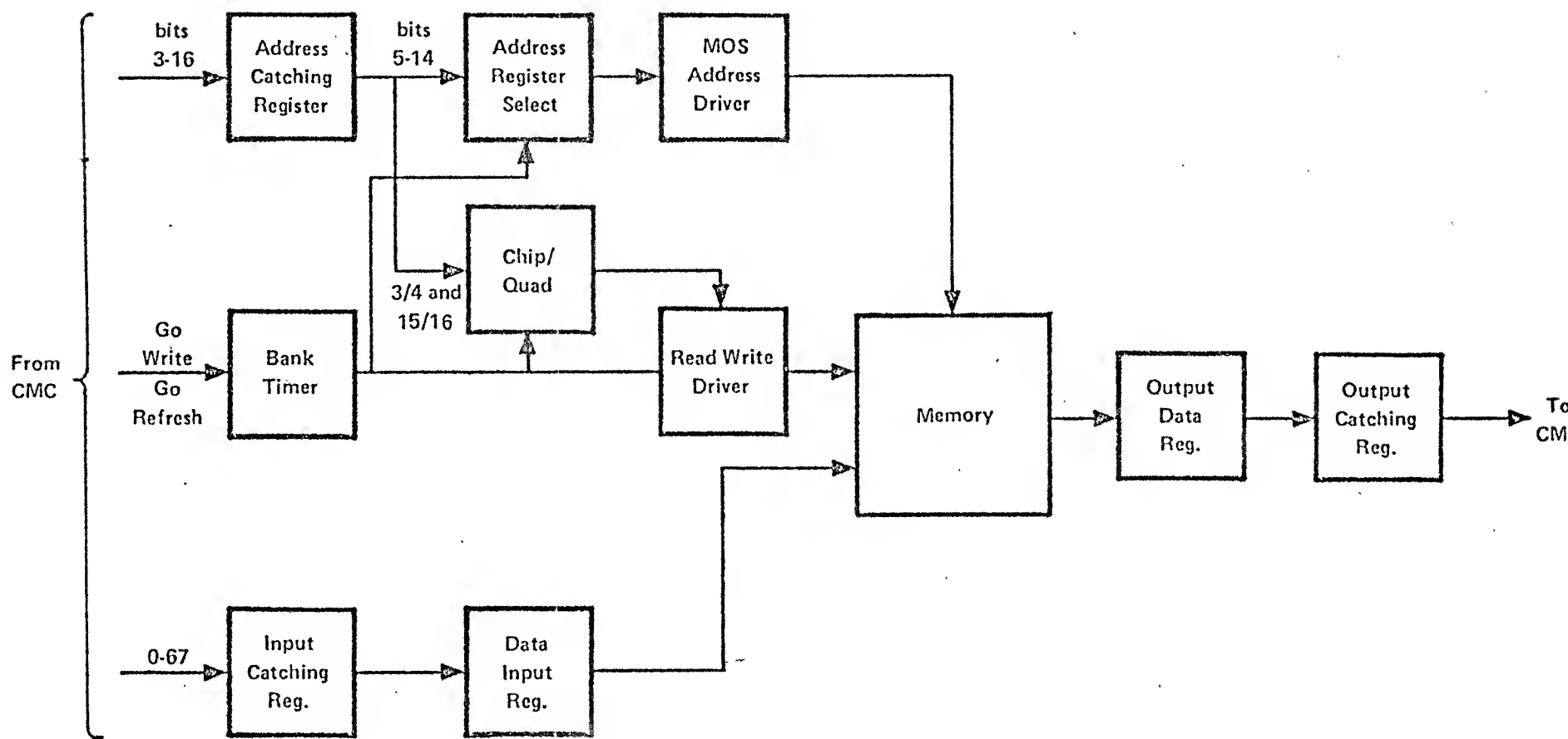
Figure 5-1-2. System Block Diagram: AA120-A (Model 175)

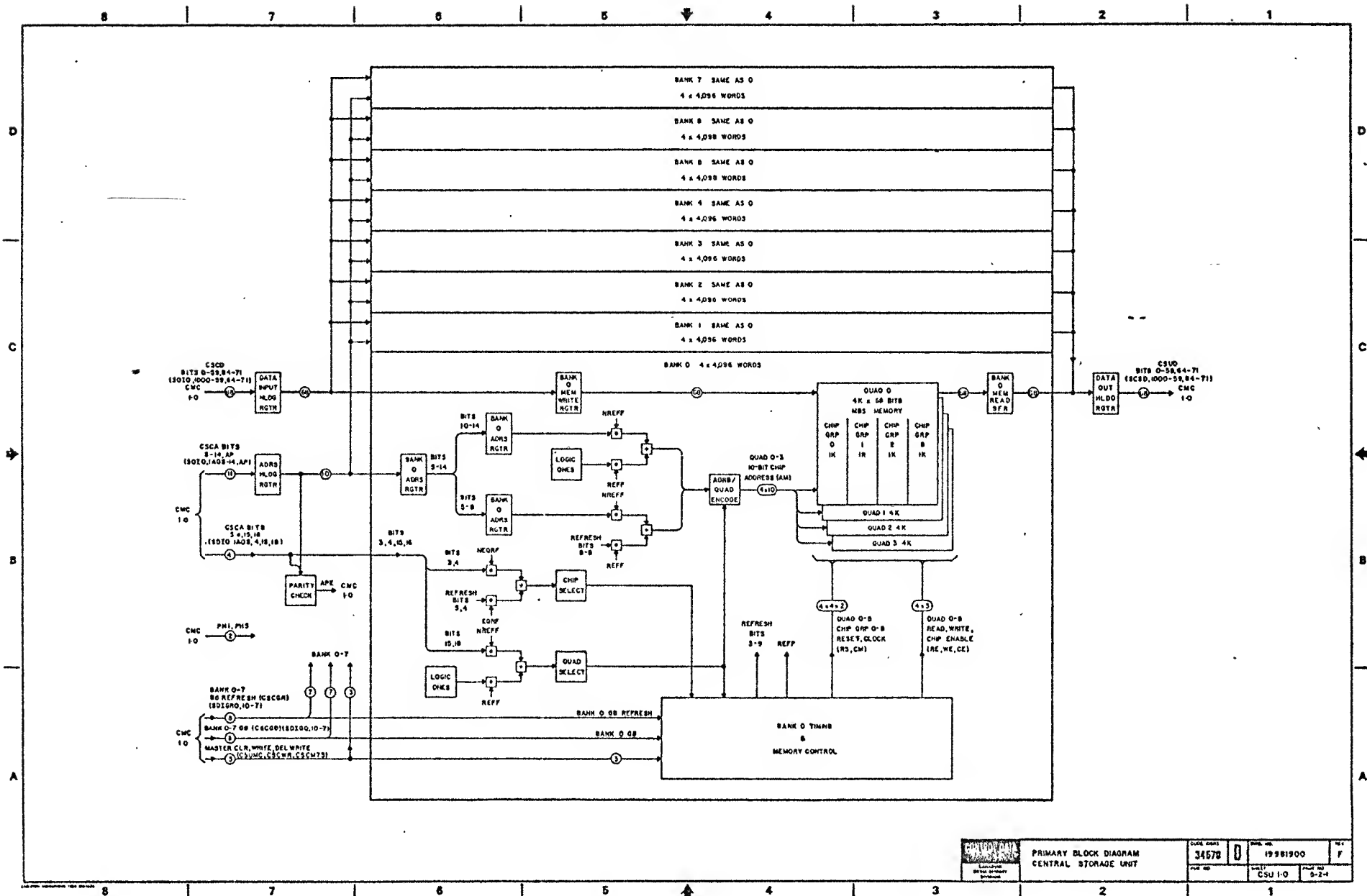
## GLOSSARY

BANK	One of 8 or 16 equal sections of memory (16 in the model 175). Each bank is divided into equal quads to a maximum of 4. Each quad contains 4096 words.
CMC	Central Memory Control.
CPU	Central Processor Unit.
CSU	Central Storage Unit.
PPS	Peripheral Processor Subsystem.
QUAD	One quarter of a bank of memory; contains 4096 words.
REFRESH	The rejuvenation of the stored charges, representing bits, within MOS memory.
RESET	The precharging of MOS memory chips for a memory reference or refresh.
SCR	Status and Control Register.
SECDED	Single Error Correction, Double Error Detection. When a single bit failure occurs in memory it is automatically corrected. When two or more bits fail the condition is simply flagged as an error, with no correction attempted.

NOTE: This glossary will be expanded at a later revision to include key signal names with explanations.

# CENTRAL STORAGE UNIT DETAILED BLOCK DIAGRAM





## PRIMARY BLOCK DIAGRAM (CSU 1.0)

### GENERAL DESCRIPTION

Each central storage unit (CSU) is divided into eight banks, each bank containing one to four quads. One quad contains 4096 words; one word contains 68 bits.

A basic CSU consists of one quadrant; one quadrant consists of one quad in each of eight banks (i.e., 8 x 4096 words). Memory size in each CSU may be increased in increments of one quadrant to a maximum of four, with a special increment of 16K per CSU for Models 172 and 175.

BASIC	- 1 QUADRANT	- 32,768 Words
OPTION	- 1½ QUADRANTS	- 49,152 (172, 175 only)
OPTION	- 2 QUADRANTS	- 65,536 Words
OPTION	- 3 QUADRANTS	- 98,304 Words
OPTION	- 4 QUADRANTS	- 131,072 Words

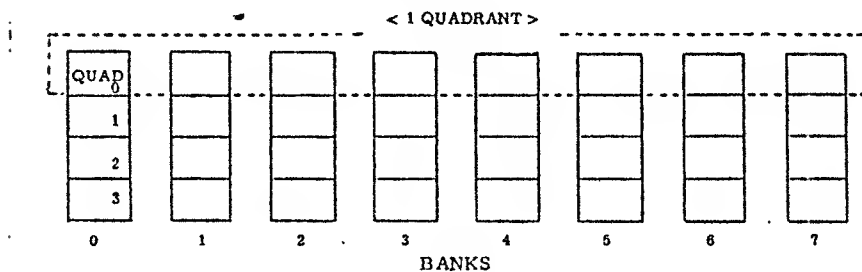


Figure 4-5-1. Bank/Quad Configuration

Each bank has its own set of address drivers, data registers and timing circuits, and operates independently from the other banks. A read or write operation is initiated when one of eight GO signals (CSC GO) is sent from CMC to a bank. This activates the bank timing circuits which generate a series of timing pulses for address selection and data strobing.

An address may be sent to the CSU every 50 nsec; however, the 400-nsec memory cycle time allows a given bank to be accessed every 400 nsec. A 400-nsec refresh cycle is initiated for each quadrant every 64 cycles of 400 nsec each.

### READ, WRITE, REFRESH CYCLES

Selection of a read or write operation is governed by the condition of CSCWR. This level is true for write and false for read.

Because information is stored in the form of a charge within the MOS chip, leakage does occur and the charge must be rejuvenated periodically. The stored charge is rejuvenated by applying two enabling levels (CLOCK and RESET) to the MOS chip. Rejuvenation takes place during the refresh cycle. A refresh cycle is initiated by CSCGR (go refresh) from the CMC.

### ADDRESSING

Each CSU requires a 15-bit address; 14 of these bits are for addressing, while the 15th is used as a parity bit. Address parity is checked by the CSU before the address is sent to all banks. When a parity error is detected, all read and write operations are inhibited, although normal bank timing is allowed to complete its current cycle. An address parity error (APE) level is sent back to the status and control register in the PPS for recording. The 15 address bits used in the CSU are received from a larger, 19-bit register in the CMC.

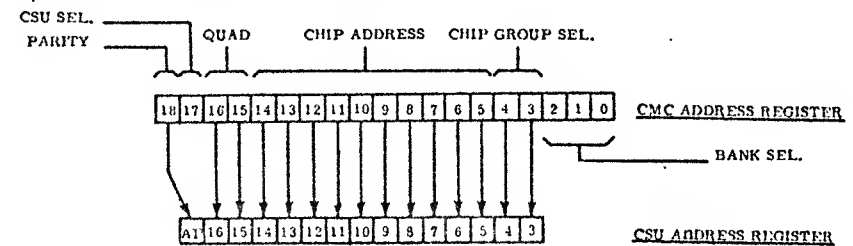


Figure 4-5-2. CMC/CSU Addressing

The first 3 bit positions in the CMC are used to select one of eight banks. Bit 17 is used to select one of two possible sets of eight banks.

### DATA

The basic CSU data word contains 60 bits of data plus 8 SECDED bits used for data error checking.

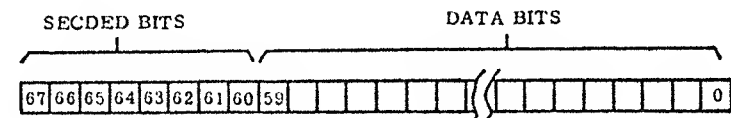


Figure 4-5-3. Data Bit Configuration

## DETAILED PAK DIAGRAM (CSU 3.0)

### CSU DATA IN/OUT PATHS

#### REGISTER COMPLEMENT

The Input/Output data paths comprise the 68-bit data input and output holding registers contained on nine KR and nine KT paks, respectively, plus the memory write and read registers combined on six EA and six EB paks for each of the eight banks.

#### DATA INPUT HOLDING REGISTER

This register, used for write operations, is spread over nine modules, with 8 bits to each module for bits 0 through 39 and 7 bits per module for bits 40 through 67.

#### MEMORY WRITE REGISTERS AND READ BUFFERS

The 68-bit output from the input holding register is distributed to the write registers for all eight banks using daisy-chained back panel wiring. During a write operation, data is clocked into one of the write registers with STRx which is derived from the corresponding GO signal sent by CMC. The output of the write register is then strobed into the DIO lines to the MOS memory. For a read operation, information in the memory data lines (DIO) is strobed with DOG and clocked into the data output holding register for transmission to CMC.

The DIO lines are common to both the write and read operations; separation of the two operations is implemented by the timing differences of data input gate (DIG) and data output gate (DOG). The latter occurs near the end of the memory access cycle.

The DIO lines are connected to the write and read register/buffers via wired ANDs. Of the 68 DIO lines for each bank, 44 are ANDed with foil connections on the EA and EB modules; the remaining 24 bits for each bank are ANDed in the back panel.

#### DATA OUTPUT HOLDING REGISTER

Read data from the read buffer is clocked into the data output holding register by the ninth T13 after the GO signal is received from CMC (T400 of the memory cycle).

## DETAILED PAK DIAGRAM (CSU 3.1)

### MOS MEMORY

#### ORGANIZATION

The MOS memory is contained on the CS and CU pak types, each containing a 5-bit (CS) or a 4-bit (CU) portion of a 68-bit word. Twelve CS paks are used to construct the first 60 bits (data portion), while two CU paks are used to construct the last 8 bits (check bit portion).

Each CS pak is organized as a 5x4 matrix comprised of 20 1x1024 MOS RAMs, as shown in figure 4-5-4. Each of the five rows can store 1 bit at  $4 \times 1024 = 4096$  locations. Each of the four columns can store a 5-bit portion of the data word at 1024 locations. CU paks are similarly arranged, holding a 4-bit portion of the 8 check bits at 1024 locations.

Each of the eight banks, in the case of the largest memory option, is divided into four quads. A quad consists of 12 CS and 2 CU paks, and is capable of storing an entire 68-bit word at 4096 locations. A quad can be further divided into four chip groups, corresponding to the four columns appearing on each of the 14 paks.

The memory stack on the DPD shows the maximum configuration for one chassis (8 banks x 4 quads). Organization of the AA107-A and AA120-A basic memories, plus increments to maximum size, are shown in figure 4-5-5.

#### DATA DISTRIBUTION

Each of the DIO lines (68 bits per bank) are daisy chained to four pak locations to cover all four quads; e.g., bank 0 bits 0-4 are applied to chassis locations C32, C31, A32, and A31 (quads 0-3, respectively).

#### ADDRESS SIGNAL DISTRIBUTION (AMxxx)

Thirty-two sets of 10-bit chip address signals are applied to the memory stack, one address for each bank/quad combination. Each 10-bit address is daisy chained to all 14 paks of the corresponding bank/quad.

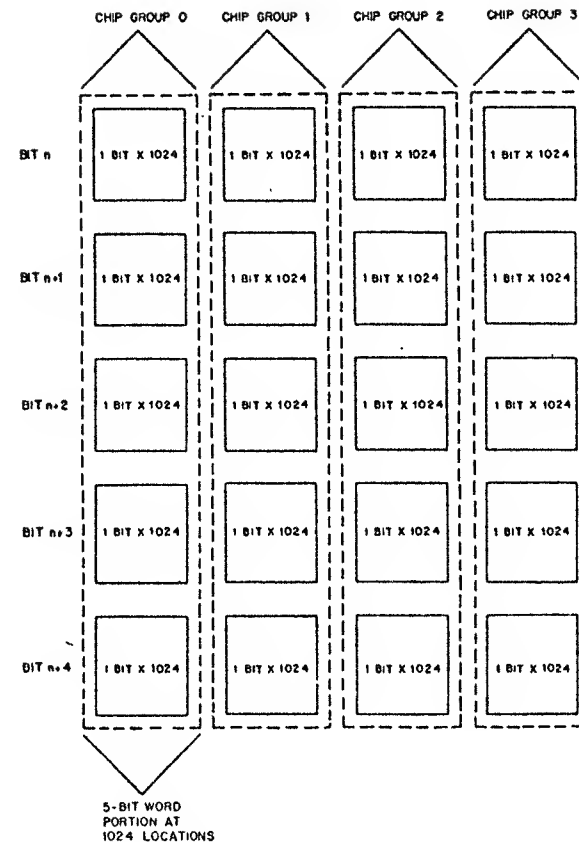


Figure 4-5-4. Typical CS Pak Memory Array

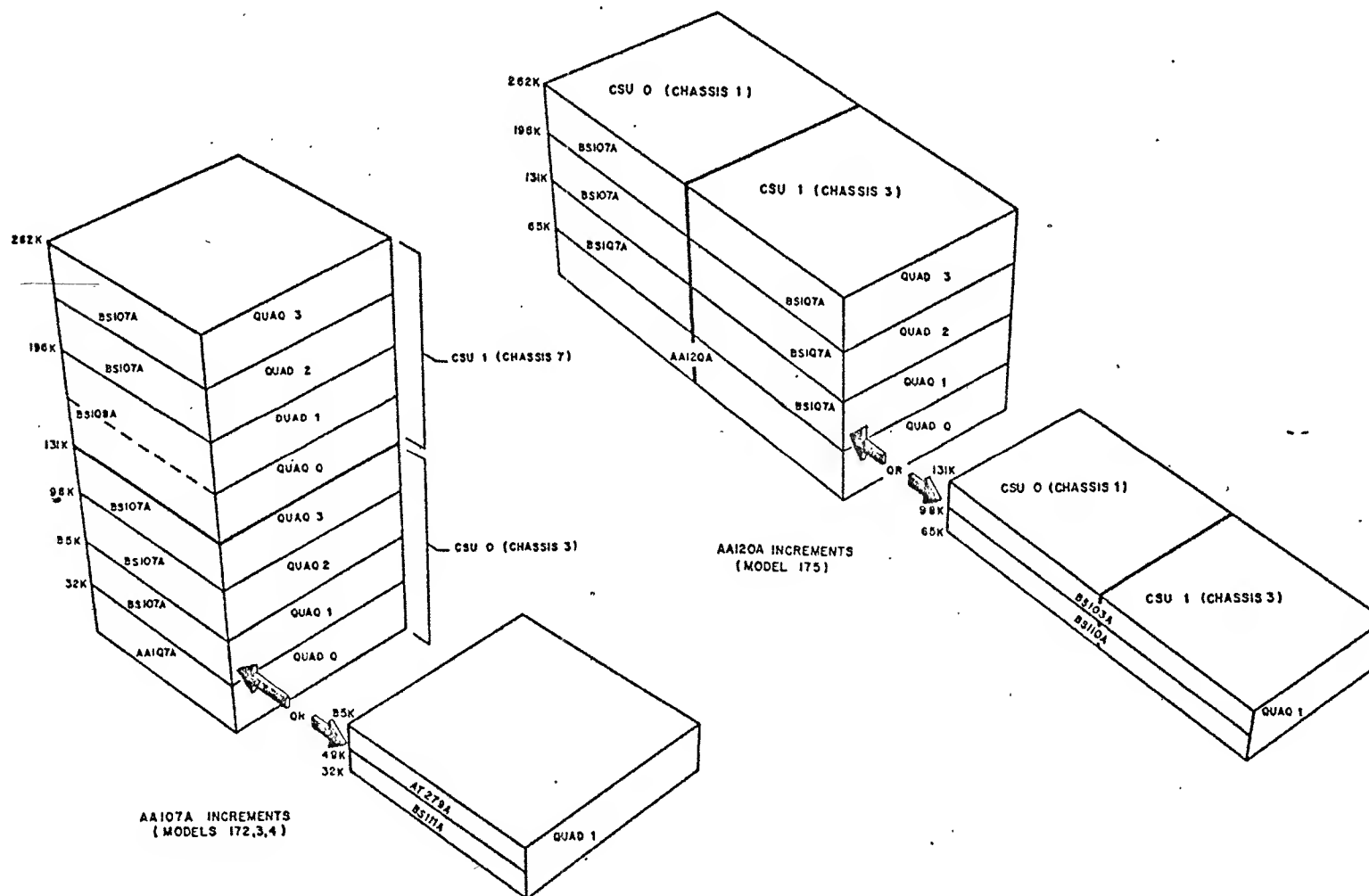


Figure 4-5-5. Organization of Memory Increments

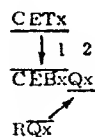
## CONTROL SIGNAL DISTRIBUTION

Control signals are distributed in three different ways:

### Chip Enable (CEBxQx)

Distribution of the chip enable signal is identical to that of the chip address. Thirty-two signals are applied to the memory stack, one for each bank/quad combination. One signal is applied to a pak.

Signal names are designated as follows:



Note	Fctn.	Range
1	Bank	0-7
2	Quad	0-3

### Write, Read Enable (WEx, REx)

One write enable signal and one read enable signal are generated for each bank. Distribution to each pak in a bank is accomplished via fanout at the source.

Signal names are designated as follows:

1  
WEx  
or REx

Note	Fctn.	Range
1	Bank	0-7

### Reset, Clock (RSxxxx, CMxxxx)

The reset and clock signals are generated as 32 pairs of logically unique signals, one pair for each bank/quad. However, because of design considerations, each signal is duplicated 3 or 5 or 7 times at the source. The reset signal is applied to the memory stack as 32 sets of 7 wires, with a distribution of two paks per wire.

The clock signals for banks 0 and 7 are supplied as 32 sets of 5 wires: 4 wires to each of three CS paks, 1 wire to two CU paks in a quad. Banks 1 through 6 have 3 wires per clock signal: 2 wires to each of five CU paks and 1 wire to the remaining two CS paks and two CU paks.

## DETAILED PAK DIAGRAMS (CSU 3.2, 3.3)

### ADDRESS PATHS

#### CENTRAL MEMORY ADDRESSING

Addressing for each of the 131,072 word locations in one CSU consists of four elements: bank selection (1 of 8); quad selection within a bank (1 of 4); chip group selection within a quad (1 of 4); word addressing within a chip group (1 of 1024).

#### BANK SELECTION

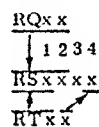
Bank selection is encoded by the CMC in the eight GO signals as determined by bits 0, 1, and 2 of the memory address presented to CMC. Control signals derived from GO are gated with all memory signals so that eight sets of data, address and control signals are presented to the memory stack (CS and CU paks).

#### QUAD AND CHIP GROUP SELECTION (RSxxxx and CMxxxx)

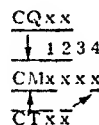
Quad and chip group selection takes place as a result of earlier multiplexing of memory address hits 3, 4 and 15, 16 on the EF paks, which generate RQ, RT, CQ and CT signals. RS signals are generated from RQ and RT; CM signals are generated from CQ and CT.

RS and CM signals are applied in pairs, 4 pairs to a pak. These 8 signals will activate a particular chip group in a particular quad.

Signal names are designated as follows:



Note	Fctn.	Range
1	Bank	0-7
2	Quad	0-3
3	Fanout	A-G
4	Chip Group	0-3

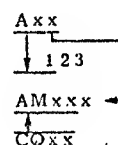


Note	Fctn.	Range
1	Bank	0-7
2	Quad	0-3
3	Fanout	A-G
4	Chip Group	0-3

#### WORD ADDRESS (AMxxx)

The 1024 word locations in each chip group are called up with a 10-bit word address which is derived from bits 5-14 of the memory address.

Signal names are designated as follows:



Note	Fctn.	Range
1	Bank	0-7
2	Quad	0-3
3	Bit	0-9

#### REFRESH ADDRESS

During a refresh cycle, the normal mode of addressing is bypassed. An address is substituted such that the 5 most significant bits of the chip address (bits 10-14) as well as the quad select bits (15, 16) are set to logic one. The remaining bits in the memory address (bits 5-9 and 3-4) are gated out by the REFF signals and the output of a 7-bit refresh counter substituted. The organization of the address is shown in figure 4-5-6.

After every 64 memory cycles (provided that ECS transfer or an exchange jump is not in progress) a refresh cycle is initiated, and all banks and their quads are activated.

Thirty-two word locations in a given chip group from all bank/quad combinations are refreshed on the first pass. A total of 128 word locations per bank will therefore be refreshed. The counter will then increment, and since the chip group is selected by the two least significant hits of the counter output, the next chip group from all bank/quad combinations will now be valid for refresh. Thirty-two corresponding word locations in the next chip group (for all bank/quad combinations) will now be refreshed. This procedure will continue until a particular block of word locations has been refreshed for the entire memory (i.e., 4 passes). The chip address will then increment and the next block of 32 word locations will be refreshed in the same manner. The procedure is shown in figure 4-5-7.

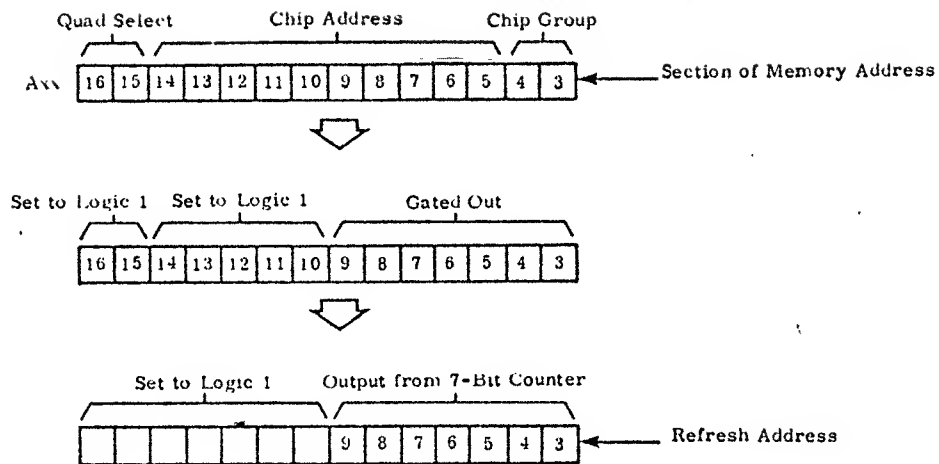
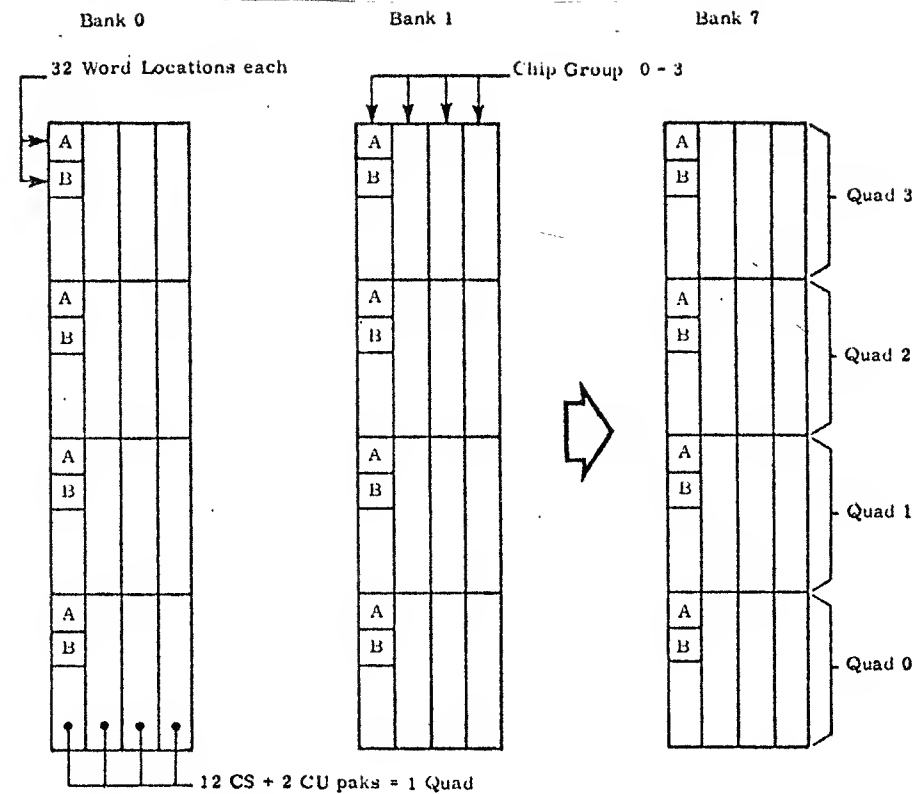


Figure 4-5-6. Refresh Addressing Scheme



Notes: A = First Refresh Cycle  
B = Second Refresh Cycle

32 such cycles required to refresh all word locations

Figure 4-5-7. Refresh Cycle

DETAILED PAK DIAGRAM (CSU 3.4)  
CSU BANK TIMING AND CONTROL

TIMING CHAIN

All three memory operations (for each bank (read, write, refresh) are controlled by a timing chain located on the EN pak. The timing chain is initiated by REON, which is derived from one of the eight GO or GO REFRESH signals sent by CMC. Once initiated, the chain is free running in response to T13.

READ/WRITE CYCLE

During a read/write operation, the EN pak generates a number of sequential control signals that provide the necessary gating for the data and address paths and memory control (see figure 4-5-8). The REON signal at T0 initiates the timing chain, and clocks the address register. The bank busy flip-flop is set, blocking the GO signals to prevent starting a new cycle before completion of the current one.

The chip enable signal (CET) is generated at T65 combined with the quad select signals to provide an enabling level for the MOS memory chips.

During a write operation, CET is combined with write to provide data input gate (DIG). This signal strobes data onto the DIO lines and also provides the write enable (true) or read enable (false) levels for the MOS memory. CET is inhibited, to block read/write operations, if an address parity error has been detected (APE flip-flop set).

The memory clock on and off (CLKON, CLKOFF) signals are tapped off the timing chain. The data output gate (DOG) is generated for all memory cycles, but used only during read operations to strobe the DIO lines.

REFRESH CONTROL

During a refresh cycle, a read/write cycle takes place at 1024 address locations simultaneously. The refresh cycle is initiated by GO REFRESH a maximum of once every 64 memory cycles (25.6  $\mu$ sec). Refresh addresses are generated by the 7-bit refresh address counter. The 5 most significant bits of the refresh address select 32 word locations in all quads in one of four chip groups, all eight banks. The 2 least significant bits select the chip. A complete refresh of 131,076 words of memory thus takes  $32 \times 4$  refresh cycles. Protection against too frequent refresh is provided by the memory disable FF. This FF is set by RFD if less than 4.8 microseconds have elapsed between two REFF signals.

CMC  
171, 172, 173, 174

## INTRODUCTION

### SYSTEM BLOCK DIAGRAM

Central memory control (CMC) forms an interface between the central storage unit(s) and the major subsystems in the computer.

These subsystems are:

Peripheral processor subsystem (PPS-0, PPS-1)  
Central processor units (CPU-0, CPU-1)  
Extended core storage coupler (ECS coupler)

Central memory control:

- permits each of the five possible units to request CM access (CSU-0; CSU-1). Input ports at the CMC will accept a 60-bit data word and an 18-bit address from each unit; output ports gate through data originating in central memory (CSU) back to the requesting unit.
- generates an 8-bit SECDED (single error correction double error detection) code that is added onto each data word to be written into CM. Codes subsequently received during a read operation are examined, and a syndrome code is generated that automatically corrects single bit parity errors. This obviates the need for a requesting unit to make another read request, thereby increasing machine efficiency.
- processes data requests every 50 ns. Requests accepted by the CMC will be permitted CM access immediately, provided that refresh operations are not underway. Simultaneous requests are dealt with according to a priority status assigned to each unit.
- controls the exchange jump package, a more specialized form of data transfer. This operation permits a 16-word package of data to be interchanged between a specified area in CM and one in either the PPS or CPU.
- controls updating operations required by CM. The CMC will block memory access during certain times such that refresh operations may occur. Data requests are handled between these periods, except during ECS requests when refresh and data access will be phased in together. Additionally, CMC controls all signals required by CM for read and write cycles.

- receives breakpoint information from the status and control register. This contains a memory address and the breakpoint code. CMC will match incoming addresses with this address and act according to instructions contained in the code.

## GLOSSARY

BANK	One of eight equal sections of memory. Each bank is divided into equal quads to a maximum of 4. Each quad contains 4096 words.
BREAKPOINT	The condition that exists when the current memory address residing in the CMC matches an address specified in the status and control register.
CEJ/MEJ SWITCH	Maintenance switch in the PPS used to control the setting of the monitor flag during an exchange jump.
CMC	Central memory control.
CPU	Central processor unit.
CSU	Central storage unit.
DEGRADE	The removal of a section of memory from the system by manipulation of switches.
ECS	Extended core storage.
ECS COUPLER	Logic used to control the transfer of data between main memory and extended core storage.
GO	Signal sent from the CMC to the CSU to initiate a memory cycle.
MONITOR	A flag used to indicate that the monitor program is running in the central processor.
PASS	An instruction having standard format but performing no useful function within the machine.
PPS	Peripheral processor subsystem.
QUAD	One quarter of a bank of memory; contains 4096 words.
REFRESH	The rejuvenation of the stored charges, representing bits, within MOS memory.
SCR	Status and control register.
SECDED	Single error correction, double error detection. When a single bit failure occurs in memory it is automatically corrected. When two or more bits fail the condition is simply flagged as an error, with no correction attempted.

NOTE: This glossary will be expanded at a later revision to include key signal names with explanations.

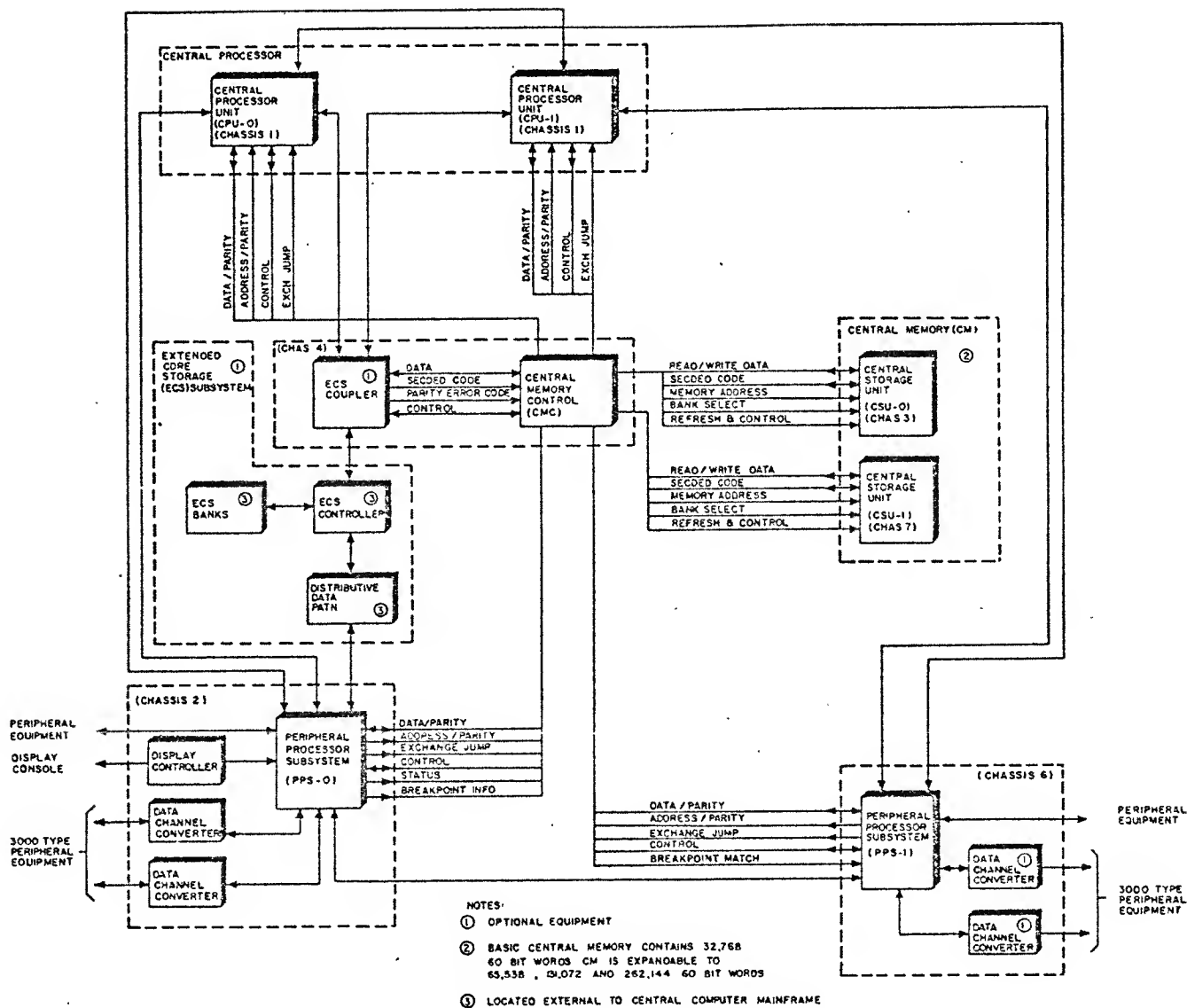


Figure 5-1-1. System Block Diagram

## PRIMARY BLOCK DIAGRAM (CMC 1.0)

Major functions of central memory control (CMC) are as follows:

### DATA EXCHANGE

The CMC allows each of the subsystems (CPU-0, 1; PPS-0, 1; ECS coupler) to communicate with the central storage units (CSU-0, CSU-1) by gating through addresses and data on a memory read or write cycle. To circumvent simultaneous requests from any of the ports, CMC assigns priorities to them to permit an orderly, sequential execution of data transfers.

CMC also contains the hardware to perform an exchange jump operation. In this procedure, a requesting port (CPU-0 or CPU-1) may exchange a 16-word package from its registers with a 16-word package stored in memory. The data package from the requesting port is held in a 16-word buffer (exchange jump write buffer), while the starting address destined for the CSU is held in the address increment and hold circuit. Data from the CPU is then deposited in the buffer, and data transferred from memory and written into the CPU. Each time this occurs, the CSU address will be incremented by one until the buffer is loaded. The address increment and hold circuit will then bring up the original starting address, and the buffer contents written into memory using the same procedure of stepping the CSU address.

### PARITY CHECK/GENERATION

#### Address

Each address word arriving at the CMC from a port is checked for possible parity errors. Should a parity error occur, the parity error flag is set, and an address parity error signal is sent to the status and control register along with the requesting unit code. A parity error signal will be sent to the requesting unit at the same time the CMC returns an accept. CMC will also block the GO signals to memory during a read or a write request. Additionally, if a memory write has been requested, the operation will be blocked, protecting the memory.

If no parity error occurs, the address will be sent to memory, along with a parity bit generated by the CMC.

#### Data

The CMC checks all five ports for possible data parity errors. Should a parity error occur, the parity error flag is set, and the requesting unit code sent to the status and control register. The lack of the address parity error signal (which is sent when an address parity

error occurs) indicates that a data parity error has occurred. A parity error signal is also sent to the requesting unit.

### SECDED GENERATION

In SECDED mode (single error correction double error detection), each data word entering a port is checked for parity errors. If none exists, then an 8-bit SECDED code is generated. Each bit of the code is derived as a parity bit based on a selected group of bits from the data word. The result is a 68-bit word sent to memory. When this word passes through the CMC during a memory read cycle, all 68 bits are gated through to the error correction logic. The 8 bits resulting from this operation are compared with the original 8 SECDED bits to produce 8 syndrome bits. Specific combinations of syndrome bits indicate either a single or a double bit failure. Single bit errors are automatically corrected; the CMC makes no attempt to correct multiple bit errors. In either case, an error report will be sent to the status and control register.

### CSU CONTROL

#### Access

Memory access required by a unit is initiated by sending a request signal to the CMC. During a 50 ns period after this signal has been received, the CMC will check parity on the data and address words, check breakpoint and CSU bank busy status, and send an accept back to the requesting unit. The CMC will then assign GO signals to the appropriate banks in the CSU to deposit or extract the required information.

#### Refresh Control

Since the CSU contains dynamic MOS devices, the CMC must periodically refresh the memory banks. Every 25.6 ns, the CMC will devote 800 ns to the memory refresh operation. During the first 400 ns of this operation, no new requests for memory access are accepted (wait period). This ensures that all current requests are processed before the memory refresh cycle begins.

However, should a refresh cycle be required during an exchange jump execution, the standard wait period will be lengthened by the time required to finish the exchange jump. During an ECS transfer no wait period is required, since the CMC will stagger memory bank access for refresh operations with that required for ECS transfer.

#### BREAKPOINT CAPABILITY

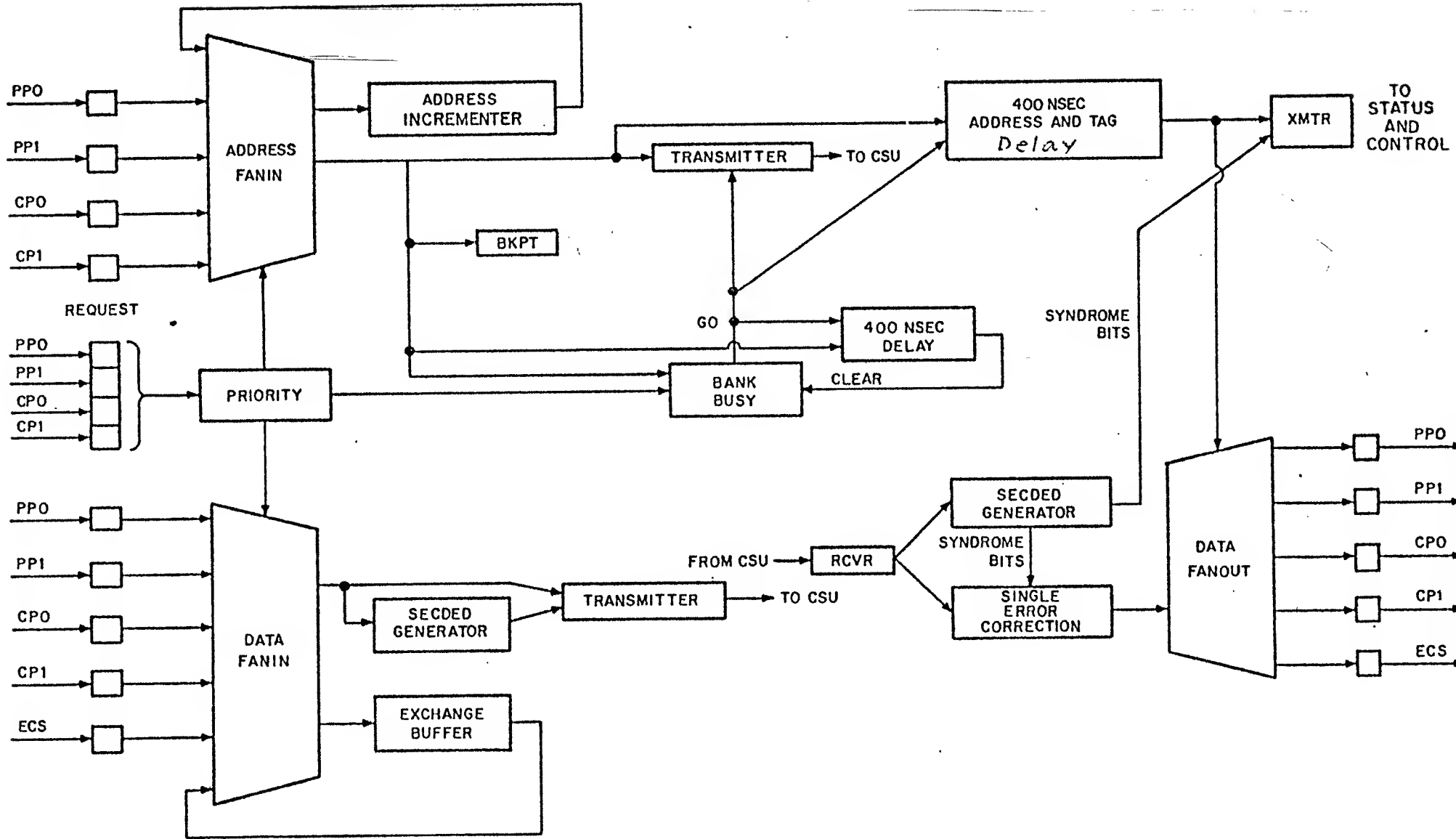
A 29-bit word is sent to the CMC from the status and control register. Part of this word (18 bits) will specify a memory address as well as a breakpoint condition code (4 bits). For each memory request from the CPU or PPS, the address will be compared with the breakpoint address. If there is a match and the requesting unit is the same as the one specified in the breakpoint condition code, and if the type of access required is also the same as specified in the code, then the breakpoint flag will be sent to the requesting unit as well as the status and control register.

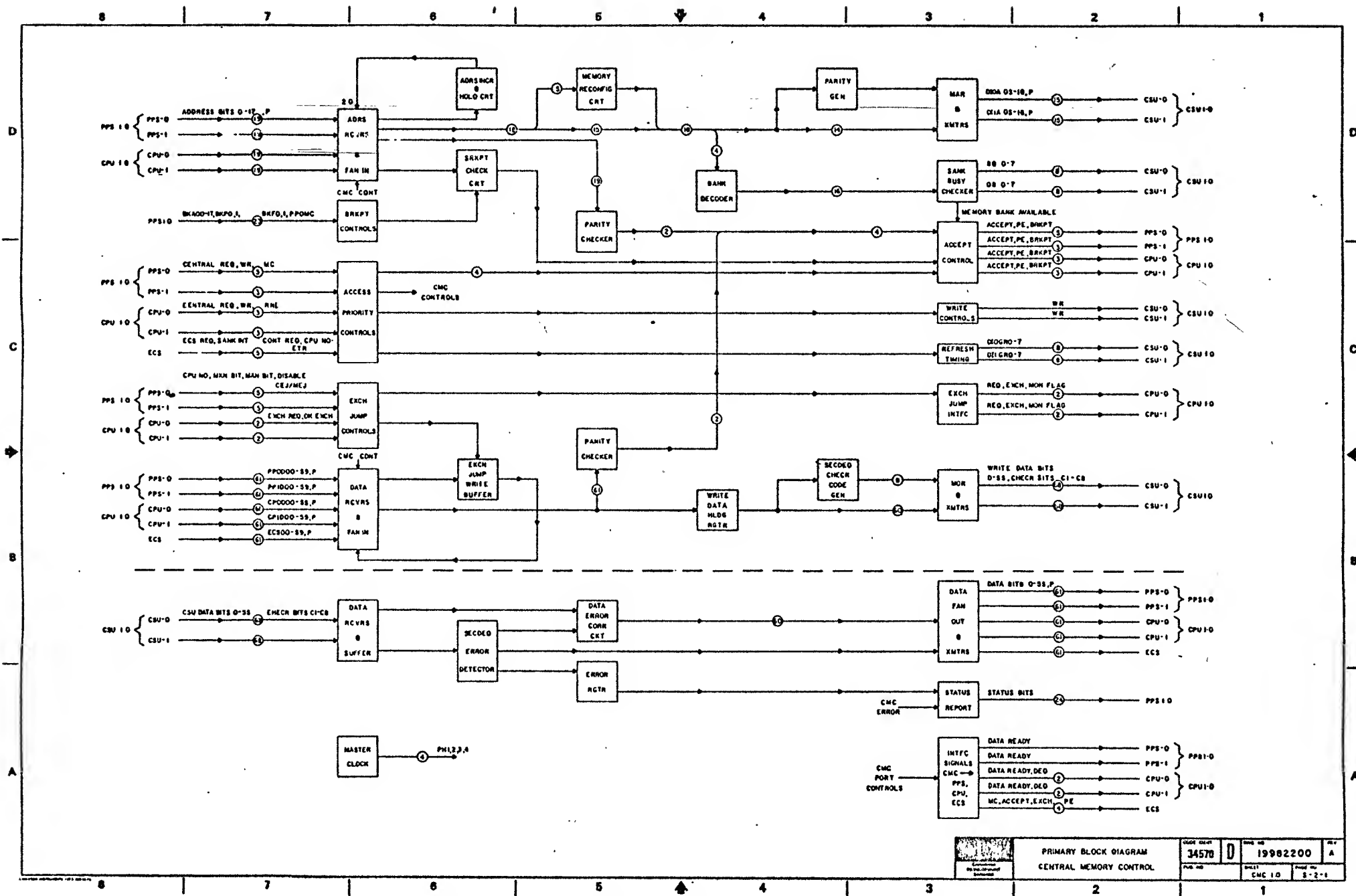
#### MEMORY DEGRADE AND RECONFIGURATION

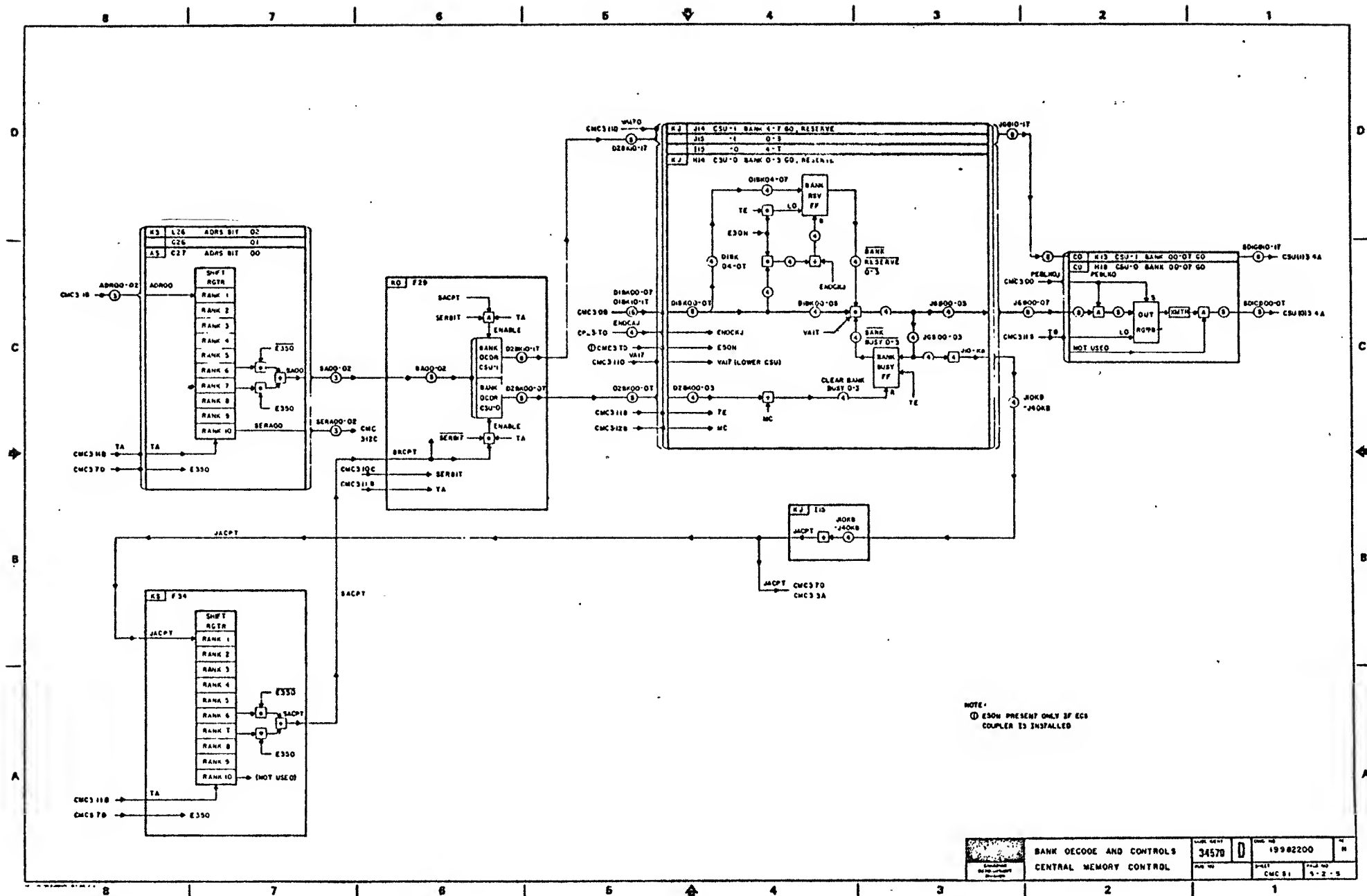
Central memory may be degraded in size with seven switches in the CMC. (Software may control the same function via the status and control register.) These switches gate out the upper 7 address bits.

Quadrants are properly reconfigured such that routine access will not be affected.

# CENTRAL MEMORY CONTROL BLOCK DIAGRAM







## DETAILED PAK DIAGRAM (CMC 3.1)

### BANK DECODE AND CONTROLS

#### SHIFT REGISTER (Address Bits 00-02)

The shift register on KS is used to store address bits 0-2 (bank address) during a memory or ECS cycle.

The bits are loaded into rank 1 and shifted one rank every 50 ns. After 300 ns (ECS transfer) or 350 ns (normal transfer), the bits are gated as SA00-02 to the bank decode circuit on KD. After 500 ns, they are sent to the status and control register (bits 048, 049, 050) to indicate the bank number if a SECDEN error occurs.

#### BANK BUSY CHECK

The GO signal for the bank selected by D1BKnn is generated and the corresponding bank busy FF is set, assuming that the bank reserve or bank busy FFs were not already set for that bank.

#### BANK BUSY FLIP-FLOP

This flip-flop is set when GO is sent to the selected bank (0-7) and reset 300 or 350 ns later by D2BKnn. This will block a second access to the same bank before the bank has completed its cycle.

#### BANK RESERVE

During an ECS transfer of 480 bits, a request for a bank will also reserve the 4th bank ahead (0 reserves 4, 1 reserves 5, etc.). The PPS is allowed to access any bank that is neither busy nor reserved. The timing relationship between PPS and ECS accesses is shown in figure 5-2-1.

#### ADDRESS PARITY

When an address parity error is detected, PAPER goes false to disable the \$DlnGO signal to the CSU, thus inhibiting the memory cycle.

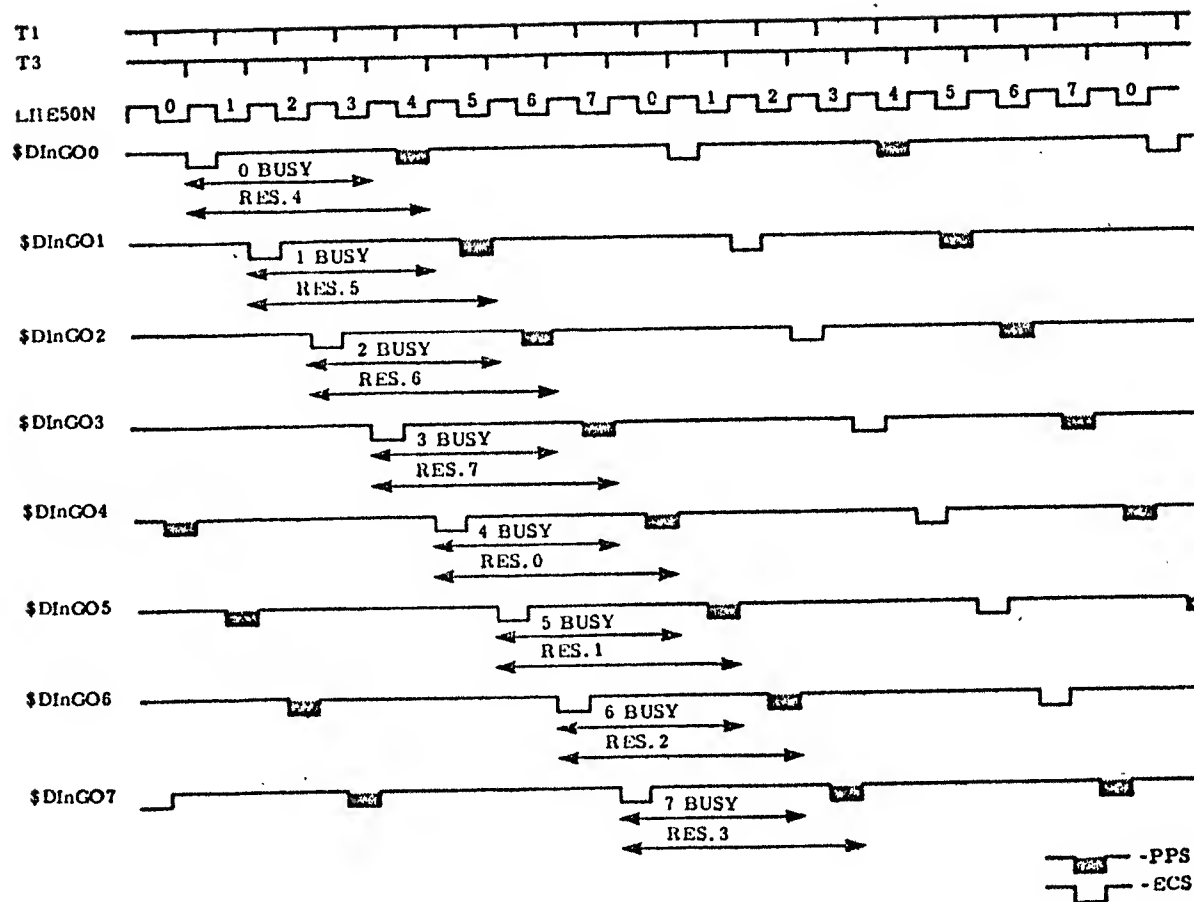


Figure 5-2-1. PPS Accesses allowed during ECS Transfer

## DETAILED PAK DIAGRAM (CMC 3.7)

### PRIORITY REQUEST CONTROLS

Priority control permits requests from the ports to be handled in sequence to prevent one unit from continuously blocking requests from other units.

Requests have the following priority:

- (1) PPS-0 XJ request
- (2) PPS-1 XJ request
- (3) CPU-0 XJ request
- (4) CPU-1 XJ request
- (5) PPS-0 normal request
- (6) PPS-1 normal request
- (7) CPU-0 normal request
- (8) CPU-1 normal request

The HA and HB levels generated by the priority control circuit are decoded on KA (CMC 3.0) to select an input port address and KW (CMC 3.8) to select data from an input port or the XJ buffer. Normal requests for priority can remain in the priority request latches when a request with a higher priority is being handled.

During an exchange jump, the priority request input selectors are blocked by KXBLKn.

Priority signals HPRI 0-3 are blocked 400 ns ahead of an ECS transfer by LHBLK from the ECS coupler.

The block enable receivers (KG) block data coming from the requesting unit until the unit is granted priority.

The TOR clocks on CMC 3.7 and 3.2 are used to control the address and data input catch registers from the requesting ports. Address and data can be latched in the catch registers by dropping the TOR clocks until priority is granted.

When breakpoint conditions have been found for the requesting unit, it is notified (GBKF) when granted priority. For further information on breakpoint, see CMC 3.8.

## DETAILED PAK DIAGRAM (CMC 3.8)

### DATA CONTROLS

#### DATA PORT SELECTION

Data input port selection levels WA, WB and WC are generated according to the unit supplying data:

Unit	WC	WB	WA
CP0	0	0	1
PP1	0	1	0
PP0	0	1	1
XJ BUFF	1	0	1
ECS	1	1	0
CP1	1	1	1

WSEL A1, B1, C1 are delayed by 500 ns to produce data ready for the requesting unit (VDRY), and also by 550 ns to gate information from the CSU into the data output register for the unit concerned (VDLH). These occur only on a ready operation. An error report is also sent to the status and control register (SEC, SECDED).

#### BREAKPOINT FEATURE

The breakpoint feature is enabled by sending two port select bits (BKP0, 1) and two function bits (BKF0, 1) along with 18 address bits from the status and control register (BKA00-17).

The 4 control bits are decoded in the CMC according to the following table:

Bit No.	117	116	115	114	Translation	X=not used
Signal	BKP1	BKP0	BKF1	BKF0		
0	0	0	X	X	Breakpoint disabled	
0	1	X	X		Breakpoint check on PP ports	
1	0	X	X		Breakpoint check on CPU ports	
1	1	X	X		Breakpoint check on PPU + CPU	
X	X	0	0		Breakpoint check on read	
X	X	0	1		Breakpoint check on write	
X	X	1	0		Breakpoint check on RNI	
X	X	1	1		Breakpoint check on any access	

When the control bits generated match existing conditions, the level WBKPT is generated and sent to the final breakpoint check circuit (3.0). If the current memory address corresponds to the breakpoint address, ABKPT (3.0) is generated and combined with WBKPT to signal a breakpoint condition (BKPT).

The requesting unit is notified (\$BKF\_) at the same time that the normal accept (\$ACP\_) is transmitted by the CMC; the SCR is sent the breakpoint flag (\$BKPT\_) along with 2 port code bits (\$HA, \$HB) and 2 access code bits (\$NWR, \$WFCT1).

During an exchange jump, both read and write operations will be sensed by the breakpoint circuitry.

When a central processor receives the breakpoint flag the following actions will occur, according to the condition of the monitor flag:

	Monitor Flag Clear	Monitor Flag Set
CEJ/ MEJ ENABLED	<p>-STORE AT RA:</p> <p>00 00 XXXXXX  ERROR P  BITS</p> <p>-XJ TO (MA),  EXECUTE PROGRAM</p> <p>-SET MONITOR FLAG</p>	<p>-STORE AT RA:</p> <p>00 00 XXXXXX  ERROR P  BITS</p> <p>-CLEAR P, STOP CPU</p>
CEJ/ MEJ DISABLED	<p>-STORE AT RA:</p> <p>00 00 XXXXXX  ERROR P + 1  BITS</p> <p>-CLEAR P</p> <p>-STOP CPU BY READING RA</p>	

## DETAILED PAK DIAGRAM (CMC 3.6)

### DATA INPUT FROM CSU

#### SECEDED

During a write operation, the CMC generates 8 SECEDED code bits that are written into memory as part of the data word.

When the same word is read from memory, 8 code bits (CSNDP1-P8) are again generated from the 60 data bits. The result is exclusive ORed with the 8 check bits from memory (CSNDC1-C8) to generate 8 syndrome bits (SY1-8) on KC. The input bit distribution for the syndrome generator is shown in note 1 on CMC 3.6.

When the total number of syndrome bits is odd, the single bit failure flag is set, and the error is corrected by inverting the data bit in error on KE. With an even number of syndrome bits, the double error flag is set and no correction is made.

#### SYNDROME BIT DECODER (KE)

The bit distribution shown on the KE pak (CMC 3.6) indicates which syndrome bits are set for a single bit failure. The syndrome bits on KE are decoded to represent the three octal numbers in the SECEDED syndrome code.

Example: Refer to CMC 3.6 and assume bit 24 is failing -

SYNDROME BITS								
58 57			56 55 54			53 52 51		
0 0			0 1 1			0 0 1		
DATA BIT NO. 24								
Syndrome Code:			0        3        1					

Referring to table 5-2-6, SECEDED Syndrome Codes, a code of 031 indicates bit 24 is failing.

#### DATA PARITY (See also CMC 3.5)

in non-SECEDED mode, during a read operation parity is propagated to the requesting unit exactly as it comes from memory and with no attempt at correction.

When SECEDED is enabled, the 8 code bits are generated by the CMC during a write operation and checked by the CMC during a read. A parity bit is generated in the CMC and transmitted to the requesting unit along with the read data.

#### MEMORY ERROR REPORT

Whenever an error occurs in the read data word, an error report is sent to the status and control register. The bank selection bits and the syndrome bits may be used to identify the failing memory module.

The bits appear in the status and control register as follows:

SCR Bit No.	Function	Signal Name
003	Mem SECEDED error	\$SECEDED (CMC 3.12)
040	SYNDROME BIT 0	\$SY01 (CMC 3.12)
041	SYNDROME BIT 1	\$SY02 (CMC 3.12)
042	SYNDROME BIT 2	\$SY03 (CMC 3.12)
043	SYNDROME BIT 3	\$SY04 (CMC 3.12)
044	SYNDROME BIT 4	\$SY05 (CMC 3.12)
045	SYNDROME BIT 5	\$SY06 (CMC 3.12)
046	SYNDROME BIT 6	\$SY07 (CMC 3.12)
047	SYNDROME BIT 7	\$SY08 (CMC 3.12)
048	SYNDROME BANK BIT 0	\$SERA0 (CMC 3.12)
049	SYNDROME BANK BIT 1	\$SERA1 (CMC 3.12)
050	SYNDROME BANK BIT 2	\$SERA2 (CMC 3.12)
051	SYNDROME QUAD BIT 0	\$SERA15 (CMC 3.12)
052	SYNDROME QUAD BIT 1	\$SERA16 (CMC 3.12)
053	SYNDROME MEM CHASSIS #	\$SERA17 (CMC 3.12)
183	DOUBLE ERROR WHEN SET	\$DED (CMC 3.12)

TABLE 5-2-6. SECDED SYNDROME CODES/CORRECTED BITS

CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT
000	NONE	040	①	100	①	160	②	200	①	240	②	300	②	340	50
001	①	041	②	101	②	161	53	201	②	241	57	301	56	341	②
002	①	042	③	102	③	162	54	202	③	242	58	302	④	342	③
003	②	043	④	103	④	163	②	203	④	243	②	303	⑤	343	④
004	③	044	⑤	104	⑤	164	40	204	⑤	244	④	304	⑥	344	⑤
005	④	045	⑥	105	⑥	165	③	205	⑥	245	③	305	⑦	345	⑥
006	⑤	046	⑦	106	⑦	166	④	206	⑦	246	④	306	⑧	346	⑦
007	⑥	047	⑧	107	⑧	167	⑤	207	⑧	247	⑤	307	⑨	347	⑧
010	⑦	050	⑨	110	⑨	150	41	210	⑨	250	43	310	48	350	⑨
011	⑧	051	⑩	111	⑩	151	③	211	⑩	251	④	311	⑩	351	26
012	⑨	052	⑪	112	⑪	152	④	212	⑪	252	⑤	312	⑪	352	⑩
013	⑩	053	⑫	113	⑫	153	⑤	213	⑫	253	⑥	313	⑫	353	⑪
014	⑪	054	⑬	114	⑬	154	⑥	214	⑬	254	⑦	314	⑬	354	⑫
015	⑫	055	⑭	115	⑭	155	⑦	215	⑭	255	⑧	315	⑭	355	⑬
016	⑬	056	⑮	116	⑮	156	⑧	216	⑮	256	⑨	316	⑮	356	⑭
017	⑭	057	⑯	117	⑯	157	⑨	217	⑯	257	⑩	317	⑯	357	⑮
020	⑮	060	⑰	120	⑰	160	42	220	⑰	260	45	320	48	360	⑰
021	⑯	061	⑱	121	⑱	161	③	221	⑱	261	④	321	⑰	361	⑱
022	⑰	062	⑲	122	⑲	162	④	222	⑲	262	⑤	322	⑱	362	⑲
023	⑱	063	⑳	123	⑳	163	⑤	223	⑳	263	⑥	323	⑳	363	⑳
024	⑲	064	㉑	124	㉑	164	⑥	224	㉑	264	⑦	324	㉑	364	㉑
025	㉑	065	㉒	125	㉒	165	⑦	225	㉒	265	⑧	325	㉒	365	㉒
026	㉒	066	㉓	126	㉓	166	⑧	226	㉓	266	⑨	326	㉓	366	㉓
027	㉓	067	㉔	127	㉔	167	⑨	227	㉔	267	⑩	327	㉔	367	㉔
030	㉔	070	㉕	130	㉕	170	③	230	㉕	270	④	330	㉕	370	㉕
031	㉕	071	㉖	131	㉖	171	④	231	㉖	271	⑤	331	㉖	371	㉖
032	㉖	072	㉗	132	㉗	172	⑤	232	㉗	272	⑥	332	㉗	372	㉗
033	㉗	073	㉘	133	㉘	173	⑥	233	㉘	273	⑦	333	㉘	373	㉘
034	㉘	074	㉙	134	㉙	174	⑦	234	㉙	274	⑧	334	㉙	374	㉙
035	㉙	075	㉚	135	㉚	175	⑧	235	㉚	275	⑨	335	㉚	375	㉚
036	㉚	076	㉛	136	㉛	176	⑨	236	㉛	276	⑩	336	㉛	376	㉛
037	㉛	077	㉜	137	㉜	177	⑩	237	㉜	277	⑪	337	㉜	377	㉜

## NOTES

- ① Syndrome code bit failed (single code bit set)
- ② Double error or multiple double error (even No. of code bits set)
- ③ Multiple single error (odd No of bit failures other than one; 3 or 5 code bits set)
- ④ Multiple error (7 code bits set)
- ⑤ Not used due to 64-bit algorithm
- 6 Syndrome codes above are octal representations of 8 syndrome code bits

## DETAILED PAK DIAGRAM (CMC 3.5)

### DATA OUTPUT TO CSU

During a write operation, the 60 data bits and 1 parity bit from the requesting units are latched by the 50-ns clock, T, in the catch register on KR. Also on KR, an additional parity bit is generated for each 8 data bits. Both the data and generated parity bits are fed to the data input selector on KB. The input selector operates under the control of WA, WB and WC control lines to gate data into the CMC as follows:

INPUT	WC	WB	WA
CP0	0	0	1
CP1	1	1	1
PP0	0	1	1
PP1	0	1	0
ECS	1	1	0
XJ Buffer	1	0	1

The 60 data bits, 1 data parity bit and 8 generated parity bits from the selected port are held for 50 ns in the write data register and then sent to the check code generator circuits on KC and KQ to generate the check code bits.

#### WRITE DATA PARITY CHECKING

The CMC checks the following data paths for parity:

- PPS-0 and PPS-1 to CMC
- ECS Coupler to CMC
- CPU-0 and CPU-1 to CMC

If a data parity error is detected in the CMC, a parity error signal is sent to the requesting unit and the following information passed to the status and control register:

1. SCR BIT 005 - CMC input parity error flag (PAPER: CMC 3.0).
2. SCR BITS 054, 055 - requesting unit code (HA-P, HB-P: CMC 3.0).

The above signals are the same as those used to indicate an address parity error with the exception of the address parity error flag, SCR BIT 139, (PAPER: KP, CMC 3.0). The absence of this line indicates a data parity error.

#### DATA CHECK CODE (SECDED)

For each 60-bit write data word, the CMC generates 8 check code bits, DIC1-C8, with each code bit representing odd parity in a selected group of data bits. The bit distribution for the check code generator is shown in note 1 on CMC 3.5. The 8 check code bits are stored in the same memory word location as the data word. (For additional information on SECDED, refer to CMC 3.6.)

A manual switch (SW #8 on the CP pak at CMN25) can enable or disable the code generator and error correction circuit, and substitute a parity bit (non-SECDED mode).

In non-SECDED mode on a write operation, parity is generated in the CMC for transmission to the CSU. If there was a data parity error on the PPS-0, PPS-1, CPU-0, CPU-1 or ECS transmission to the CMC, parity is regenerated.

In SECDED mode on a write operation, data parity is checked at the requesting unit's input port in the CMC. The 8 SECDED code bits are then generated for transmission to CM.

## DETAILED PAK DIAGRAM (CMC 3.9)

### REFRESH CONTROL I

#### INTRODUCTION

Refresh control in the CMC determines when a refresh of central memory may be initiated and, if so, whether it should be carried out as a burst or sequential refresh.

A burst refresh will simultaneously update 32 word locations in a given area of central memory in all eight banks. During the time required to do this, all memory access requests will be blocked.

A sequential refresh, carried out during ECS operations, permits ECS access and refresh to occur simultaneously by staggering the two operations across banks.

#### REFRESH COUNTER

A modulo 128 counter on the LA pak running at a 100-ns pulse rate outputs signals as shown in the pak diagram. Two of them, LAMK10 and LAMK20 form a modulo 3 counter to permit triggering on portions of the basic 400-ns clock rate needed for refresh. These signals are fed into a programmable network on the KY pak, whose output is determined by combinations of RFMHO and RFMLO from the status and control register. These allow a fast or slow refresh margin in addition to the normal one. In normal mode, Y62OR3 will be up on the leading edge of the 62nd 400-ns pulse, and last for 800-ns. Similarly, in slow mode, Y62OR3 will be up on the leading edge of the 78th 400-ns pulse.

#### REFRESH SEQUENCING

This signal (Y62OR3) is fed into the refresh sequence, along with LAMK10 and LAMK20. The outputs of this network (KY62, YRK623 and YPRES) are shown in the timing diagram. Note that a normal count is shown. These three signals will be output depending on when Y62OR3 goes high. However, relative timing between the signals will remain fixed.

When KY62 goes high and LSX is high, indicating that an exchange jump has already been initiated, the inhibit refresh FF will be set. This will lock the counter on the LA pak until the exchange jump has been completed, at which point refresh will begin again. When YRK623 goes high and NBLKRF is valid (Block Normal Refresh), a burst refresh will be performed when the burst refresh FF is set, producing YGNR.

YPRES will go high at the end of the 800-ns refresh period. This signal will reset the counter on the refresh counter back to zero.

## DETAILED PAK DIAGRAM (CMC 3.10)

### REFRESH CONTROL II

#### DETERMINATION OF SEQUENTIAL OR BURST REFRESH

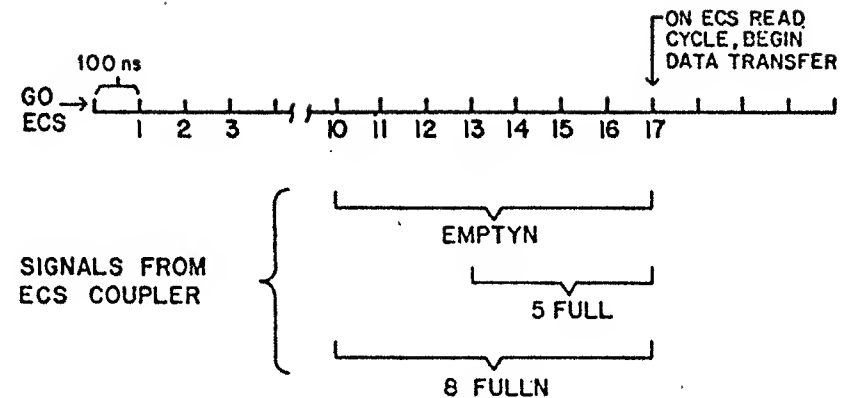
During ECS transfer, a burst refresh may be inhibited and a sequential refresh substituted. During an ECS read operation LGGOEC will be valid when the GO central FF is set. Due to the increased time (compared with an ECS write operation) required to disassemble an 8-word block from the ECS bays, signals destined for the CMC (i.e., EMPTYN, 5 FULL, 8 FULLN) are triggered from various ranks of a 17-rank shift register initiated by the ECS GO signal (see ECS Coupler Manual, CYBER 170 Models 172/173/174). This is shown schematically in figure 5-2-4. As the GO signals step through the shift register at 100-ns intervals, the three CMC signals shown in the diagram inform the sequential refresh system of time available before the ECS actually begins to transfer data which is to be written into central memory. As can be seen from the diagram and the gating on the IIE pak, when EMPTYN and 8 FULLN are true, and an ECS read operation is being done, the inhibit refresh (INHREF) will be false, and therefore a burst refresh can be done (see CMC 3.9) due to a spare time slot before ECS data transfer.

#### SEQUENTIAL REFRESH CONDITIONS

The sequential refresh FF will be loaded under the following conditions:

- during an ECS write condition, at a time determined by the outputs of the refresh counter (see CMC 3.9). LGGOEC is the output of a FF in the ECS coupler which will be valid for the duration of an ECS transfer.
- during an ECS read operation, and both LGGOEC and 5 FULL are true. 5 FULL indicates that only 300 ns remain before the first ECS data transfer. The sequential refresh FF will be loaded then according to the outputs of the refresh counter (i.e., YRK623).

When the sequential refresh FF is set, the enabling signals S1SRCT and S2SRCT are sent to the delay counter on the LA pak. These will preset the counter to zero. Note that it is a modulo 16 counter. The counter will begin to increment on the clock at 50-ns intervals. SRTION, the inverse of the least significant bit, will now perform two functions.



SIGNAL	STATUS	
	T	F
EMPTYN	Any of Ranks 10-17 full	Ranks 10-17 all empty
5 FULL	Ranks 13-17 all full	Any of Ranks 13-17 empty
8 FULLN	Any of ranks 13-17 empty	Ranks 10-17 all full

Figure 5-2-4. Refresh Signals from ECS Coupler

- on an even count, as the counter steps up to 16, SRTION will be high, enabling the decoder on the KX pak. The refresh bank decoder on the LA pak will contain the current bank address (i. e., ADR00-02) which is fed into the decoder as RFRB 0-2.
- on an odd count, SRTION will be low, driving STRONN on the KX pak high. This in turn will enable the refresh bank counter on the LA pak. This counter is constantly being fed with the running bank address (ADR 00-02) when E50N is being driven. (E50N is a 50-ns clock active during ECS transfer.) STRONN will then strobe the current bank address into the refresh bank counter. While this is being done, ECS transfer is taking place. The CMC therefore dedicates alternate 50-ns time slots to ECS and refresh, as shown in figure 5-2-5.

When the delay counter reaches a count of 16, the carry line, SRCR, goes high and feeds into the HE pak. A delay shift register will output signals (RK1-4) to reset the sequential refresh FF, and the refresh time FF (CMC 3.9). If an accept (EACP) has been received from ECS during refresh, which would have set the lost accept FF, RK4 will gate a force accept (FORREQ) to the coupler to begin that operation.

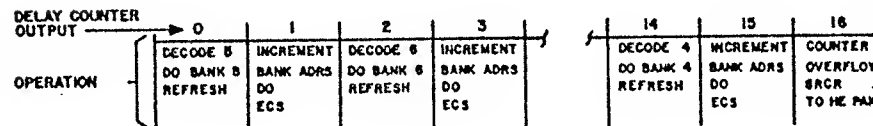


Figure 5-2-5. Dedication of Operations during ECS Transfer

# DETAILED PAK DIAGRAM (CMC 3.11) CLOCK DISTRIBUTION AND MEMORY DEGRADE

The CMC receives two clocks, T1 and T3, from the master clock generator. T1 is 25 ns wide and spaced 100 ns apart. T3 is similar to T1 but 50 ns out of phase. (See figure 5-2-6.) T1 and T3 are received by the AB paks at M25 and F32, respectively, and passed on to the AA paks to be fanned out to the rest of the chassis.

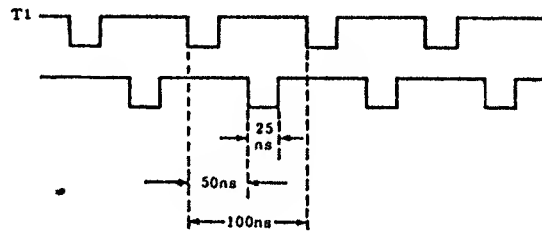


Figure 5-2- 6 . T1 and T3 Clocks

Some AA paks produce clocks only when enabled by VDHLC0, 1 or VDHLPO,1. These are used to gate data into the data output register for the unit concerned (PP0, or CPU-0,1).

The nongated clocks developed on the AA paks are 50-ns clocks and are all referred to as T throughout the CMC detailed pak diagrams.

Each AA pak has a variable delay line to adjust pulse width.

## MEMORY DEGRADE/RECONFIGURE CONTROL

Memory may be degraded in the following increments:

262K	to	196K	2ND CSU ONLY
196K	to	131K	
131K	to	98K	
98K	to	65K	
65K	to	32K	
49K	to	32K	

Figure 5-2- 7. Memory Degradation

Degradation is accomplished by manipulation of address bits 14, 15, 16 and 17 using switches 1 - 4 on the memory degrade switch paks. Individual quadrants may be removed by changing switches 5, 6 and 7, according to table 5-2-8. Degradation/reconfiguration may be accomplished alternatively by software manipulation of SCR bits 176 - 183 inclusive.

Memory reconfiguration must be accompanied by a degrade. The KZ pak changes the address of the quadrant to be removed so that it falls within the degraded section.

Example:

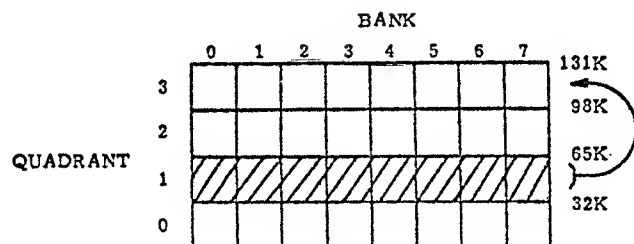


Figure 5-2-8. Memory Reconfiguration

If quadrant 1 is to be removed, memory is reconfigured to 98K and quadrant 3 is accessed in place of quadrant 1.

Memory degrade switches are located on switch paks, type 3CP0, at locations CMN24 and CMN25.

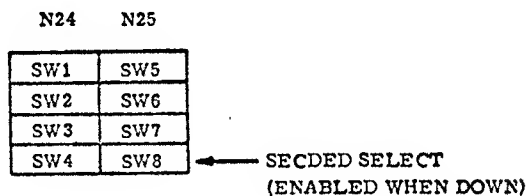


Figure 5-2-9. Memory Degrade Switches

TABLE 5-2-8. MEMORY DEGRADE SWITCH POSITIONS

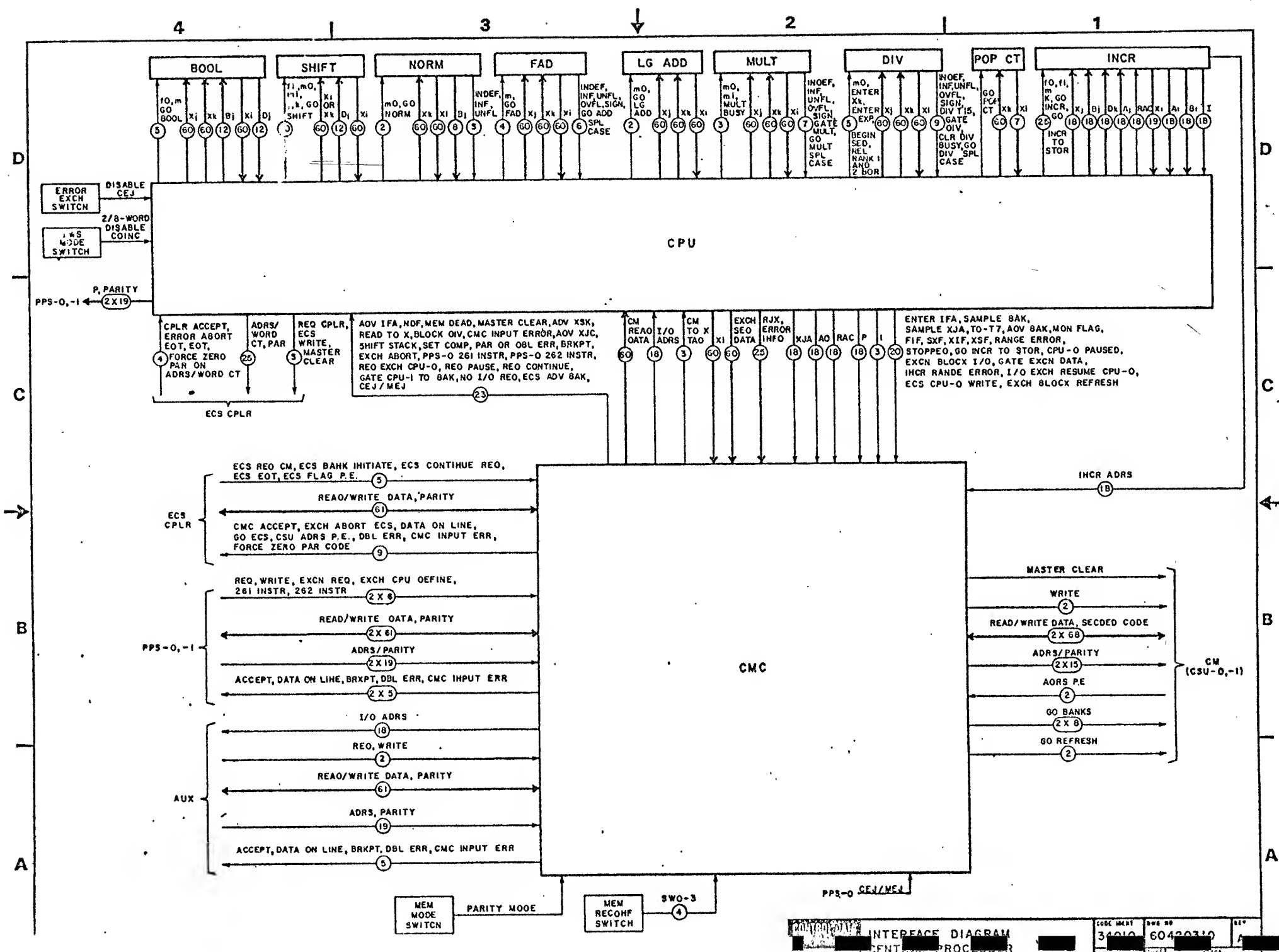
MEMORY SIZE	ADDRESS RANGE	NORMAL	DEGRADED														
		SWITCHES 1 2 3 4 5 6 7	BAD QUADRANT	SWITCHES 1 2 3 4 5 6 7													
262K	0→777777	DDDDDDDD	CSU-0	{ 00 01 10 11 00 01 10 11	DDD UUUU DDD UUUD DDD UUDU DDD UDDD DDD UDUU DDD UDDU DDD UDDU DDD UDDD	CSU 1 ONLY											
196K	0→577777	DDDUDDDD	CSU 0	{ 00 01 10 11	UDD DUUU UDD DUUD UDD DUDU UDD DDDD												
				CSU 1	{ 00 01 10 11		UDD DDUU UDD DDUD UDD DDDU UDD DDDD										
					CSU 0		{ 00 01 10 11	UDD DUUU UDD DUUD UDD DUDU UDD DDDD									
							CSU 1	{ 00 01 10 11	UDD DDUU UDD DDUD UDD DDDU UDD DDDD								
			131K					0→377777	UDDDDDDD		00 01 10 11	UDD UUUU UDD UUUD UDD UUDU UDD UDDD					
				98K							0→277777	UDDUDDDD		00 01 10	UUD DUUU UUD DUUD UUD DUDU		
					65K									0→177777	UUDDDDDD		00 01
							49K										0→137777
32K	0→077777	UUUDDDDD						NO DEGRADE									

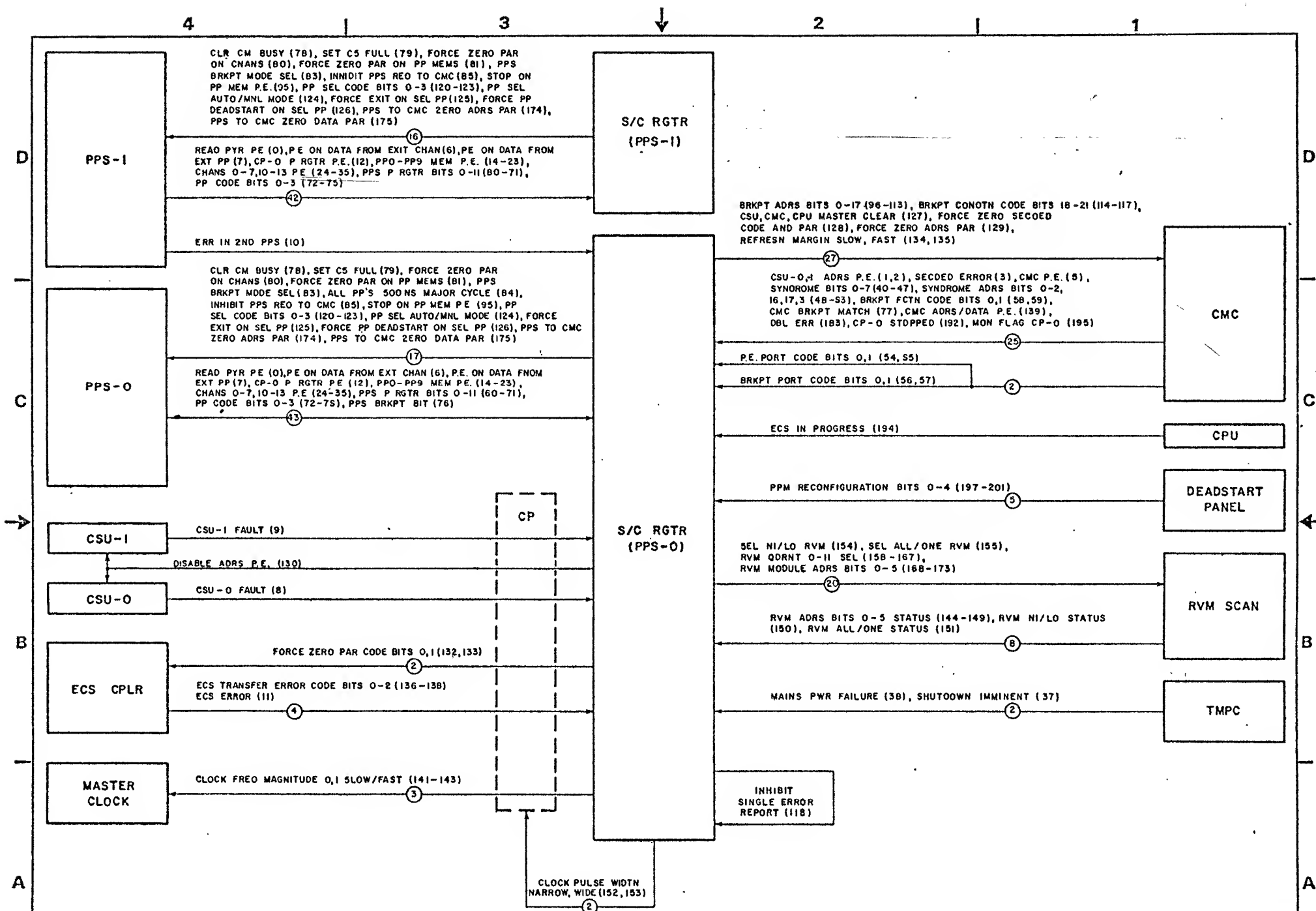
## GLOSSARY

APF Advance P flag  
ASF Advance stack flag  
BAK Block address counter  
CEJ Central exchange jump  
CIW Current instruction word  
CM Central memory  
CMC Central memory control  
CP Central processor  
CPU Central processing unit  
CSU Central storage unit  
DPXF Delayed program exit flag  
ECS Extended core storage  
EM Exit mode  
FAD Floating add  
F1F Fetch 1 (word) flag  
F2F Fetch 2 (words) flag  
FLC Field length - central memory  
FLE Field length - extended core storage  
GJF Go jump flag  
IAS Instruction address stack  
IFA Instruction fetch address  
IWS Instruction word stack  
JCF Jump completed flag  
JOF Jump out (of stack) flag  
MA Monitor address

MEJ Monitor exchange jump  
MF Monitor flag  
M1F Marker 1 (word) flag  
M2F Marker 2 (words) flag  
NBF No backup flag  
NSA Next stack address  
OSF Out (of) stack flag  
PK Parcel count  
PPS Peripheral processor subsystem  
PXF Program exit flag  
RAC Reference address - central memory  
RAE Reference address - extended core storage  
RJF Return jump flag  
RJX Return jump exit (address)  
RVM Reference voltage margin  
SAS Storage address stack  
SECDED Single-error correction double-error detection  
SWS Storage word stack  
SWA-SWD Storage word (stack - rank) A-D  
SXF Store exit flag  
XIF Exchange issue flag  
XJA Exchange jump address  
XJC Exchange jump count  
XSF Exchange sequence flag  
XSK Exchange sequence counter

CMC  
175





## CENTRAL MEMORY CONTROL

CMC provides an interface between CM and the five requesting ports (PPS-0, PPS-1, auxiliary, ECS, and CPU). All CMC interface connections (except CPU) are made with interbay cables, 156-pin connectors, I/O line drivers, and I/O line receivers. CMC and CPU are located in the same bay; therefore, interface connections are made with twisted-pair wiring. The primary function of CMC is to provide and control address and write data paths to CM and read data paths from CM. In addition, CMC:

1. Determines priority
2. Controls CM refresh
3. Controls CM reconfiguration
4. Performs breakpoint check
5. Corrects single-bit errors
6. Detects double-bit errors
7. Detects transmission parity errors

### ADDRESS PATHS

The CMC address paths consist of address selection, storage address stack (SAS), breakpoint check, and address distribution.

### ADDRESS SELECTION

The SAS receives addresses from four sources: instruction fetch address (IFA) adder, block address counter (BAK) register, I/O address selection network, and increment unit.

The IFA circuit contains two operand selection networks, two operand holding registers, and a two-stage adder. The IFA adder outputs the absolute address for the next instruction to be requested from CM. Normally, the IFA is advanced by one each time the adder output enters SAS. For branch out of stack or return jump instructions, the adder outputs  $P + RAC$ .

The BAK circuit provides addresses for the SAS during exchange jump sequences and ECS block copies. For exchange jump sequences, the initial address is the content of XJA. For ECS block copies, the initial CM address is  $A0 + RAC$ . The BAK circuit counts through the remaining addresses.

The I/O address selection network includes a separate address holding register for each I/O port (PPS-0, PPS-1, and auxiliary). The register with priority is selected to send an address to the SAS and to the XJA register. The selected address is partly checked after it receives priority to enter the SAS or XJA register. A parity error signal is sent to the status and control register and back to the requesting port. A parity error port code is also sent to the status and control register.

During 50 through 57 instructions (1 is not 0), the increment result plus RAC is sent from the increment unit to the SAS. Since there is no holding register for increment addresses in CMC, the increment instructions do not issue until the SAS is available to accept the address.

### STORAGE ADDRESS STACK

The SAS is a buffer for addresses arriving at CMC. Two registers in the SAS are referred to as rank A and rank B. These registers hold a CM address and control information. SAS allows CM requests to be accepted without the required bank being free.

If no CM bank conflicts occur, an address enters SAS rank A in any given 50-nanosecond clock period and leaves in the following 50-nanosecond clock period. Rank B is not used even if a new address enters rank A every clock period.

When a CM bank conflict occurs, the address in rank A is held until the conflict is resolved. If another address arrives at SAS during this time, it enters rank B. When this occurs, no more addresses enter the SAS until the backup condition is resolved. Once a backup condition has occurred, no new addresses

are accepted by SAS until the addresses in both ranks have been sent to CM. Addresses leave SAS in the same order that they entered. (Rank A empties first; rank B empties last.)

#### BREAKPOINT CHECK

As addresses enter SAS, they are compared with the breakpoint address from the status and control register (PPS-0). If the addresses match, a breakpoint signal is conditionally sent to the status and control register and to the requesting port. A breakpoint condition code in the status and control register determines what type of CM references causes a breakpoint signal to be generated.

#### ADDRESS DISTRIBUTION

The location of each word in CM is identified by an 18-bit address. Bits 0 through 3 are used to select one of 16 banks. Bit 3 also selects one of the two central storage units (CSUs). Bits 4 through 15 select one of 32,768 words in a CSU quadrant. Bits 16 and 17 select one of four CSU quadrants.

Bank selection is based upon the condition of address bits 0 through 3. Banks 0 through 7 are located in CSU-0. If address bit 3 is clear, one of banks 0 through 7 is selected. If address bit 3 is set, one of banks 10 through 17 in CSU-1 is selected.

The 16 bank busy registers correspond to the 16 banks in CM. These registers are used to keep track of which banks are currently active with a CM reference. Address bits 0 through 3 determine which bank busy register is set and which bank receives a go bank signal to start a CM reference. If the bank is already busy, the go bank signal is not generated until the bank is free.

The 16 bank reserve registers correspond to the 16 banks in CM. These registers ensure that a bank required for an ECS transfer will be free. This allows the block copy to proceed at a rate of one word per 100 nanoseconds. Bank reserve registers are set for banks N+1, N+2, N+3, and N+4 when bank N busy register is set by an ECS transfer. Bank reserve registers are cleared when the corresponding bank busy register is set or when a record gap exists in the ECS transfer. For non-ECS references, reserved banks appear to be busy and cannot be accessed until they are not busy and not reserved.

The SAS transmits address bits 4 through 17 (plus parity bit) to both CPUs when the go bank signal is transmitted. Address bits 15 through 17 also pass through a memory reconfiguration circuit.

The address delay chain receives address bits 0 through 3, 16, and 17 from the SAS. These bits are delayed and sent to the status and control register along with syndrome code bits 0 through 7 from the SECDED check network. This information can then be interpreted to isolate CM failures.

#### WRITE DATA PATHS

CM transmits a 60-bit data word (plus 8-bit SECDED code) to both CSUs of CM. Two write data paths exist. One path is for I/O write data from PPS-0, PPS-1, ECS, and auxiliary ports, and the other path is for write data from the CPU. The two paths are merged in the storage word stack (SWS) before the data is sent to CM.

#### I/O WRITE DATA

The I/O write data path consists of four write data holding registers, a selection network, parity checking network, and SWS banks B through D (SWB through SWD). I/O write data is merged with CPU write data at the input to SWC.

Each I/O port has a separate 60-bit (plus parity bit) write data holding register. Each register functions independently. Data enters and is held in the registers until selected for use. The selected data is parity-checked and gated to the SWB I/O register which is used to hold I/O data during an SAS backup condition. An I/O data parity error signal is sent to the status and control register and to the requesting port. A parity error port code is also sent to the status and control register.

#### CPU WRITE DATA

The CPU write data path consists of SWS ranks A through D (SWA through SWD). The SWS is a buffer area for 60-bit words which are to be written into CM.

SWA is the first rank of the SWS. This register is normally cleared and entered with XI during each 25-nanosecond clock period. This data is shifted through the four ranks of SWS and written into CM only when an increment instruction is issued which requires a CM reference. In all other cases, the data is shifted through the ranks and discarded. This normal mode of operation is altered if an exchange jump request is received.

During the first half of an exchange sequence, the XI register data is blocked and the first eight 60-bit words of the exchange package (A, B, and support register data) are shifted through the SWS into CM. The last eight words of the exchange package consist of X register data which is shifted through the SWS in the normal mode of operation.

SWB is the second rank of SWS. It consists of two 60-bit registers which are normally cleared and entered with write data during each clock period. Only one of the two SWB registers can contain valid write data at a given time. The SWB I/O register was described earlier. The SWB register is normally cleared and entered with the content of SWA during each 50-nanosecond clock period. This normal mode of operation is altered in the event of a CM conflict or when a return jump instruction is issued.

In the event of an SAS backup, the data is held in SWB until the conflict that caused the backup has been resolved.

During a return jump sequence, the SWA is blocked and RJX enters bit positions 30 through 47 of SWB. If the return jump sequence is caused by an error exit condition, bits 48 through 53 contain error information and bits 54 through 59 are zeros. If a return jump instruction caused the sequence, bits 48 through 59 contain a 0400 jump instruction.

SWC is the third rank of SWS. This register is normally cleared and entered with SWB or SWB I/O each 25-nanosecond clock period. This normal mode of operation is altered in the event of a CM conflict. When a conflict occurs, the data is held in SWC until the conflict has been resolved.

Eight SECDED code bits are generated from the data in SWC. If parity mode is selected, a single odd parity bit is generated. The SECDED code generator is described later.

SWD is the fourth and last rank of the SWS. This register is unconditionally cleared and entered with SWC and the eight SECDED code bits during every 50-nanosecond clock period. The 68-bit word is transmitted to both CSU-0 and CSU-1 during every clock period.

#### READ DATA PATH

The read data path consists of the read data holding register, the single-error correction double-error detection (SECDED) network, and the I/O read data holding registers.

#### READ DATA HOLDING REGISTER

This register receives a 60-bit (plus 8-bit SECDED code) read data word from CM during every clock period. Only one of the two CSUs of CM sends valid data during any given clock period. If address bit 3 is set, CSU-1 data enters the holding register. If clear, CSU-0 data enters. Read data enters the holding register 500 nanoseconds after a go bank signal is sent to CM.

## SECODED NETWORK

The read data path provides SECODED or odd parity checking on data read from CM. A switch located on chassis 5 selects the SECODED or parity mode of operation. SECODED is the normal mode. The SECODED network is described in detail under read data paths (CMC 2.2).

## READ DATA DISTRIBUTION

Read data from the SECODED network is sent to the I/O read data holding registers, the IWS, the X registers, and the A, B, and support registers. During the first eight words of an exchange read sequence, data is sent to the A, B, and support registers.

Each I/O port (PPS-0, PPS-1, ECS, and auxiliary) have a separate read data holding register. Data (plus parity) enters and remains in a register until new data is entered in the same register. The output of each register is transmitted to its respective port.

## REFRESH CONTROL

Refresh is in effect a CM read operation. Therefore, the banks to be refreshed must be free. All banks are normally refreshed every 25.6 microseconds. A go refresh signal is sent to all 16 banks in CM. Each signal causes a refresh reference to be started in the corresponding bank. The signals are sent in two groups of eight so that all eight banks of a CSU start a refresh reference at the same time. All 16 banks can start at the same time (normal mode) or at two different times (ECS mode). Go refresh is never sent within 400 nanoseconds of a go bank signal to the same CSU.

## MEMORY PRIORITY AND DESTINATION CONTROL

Memory priority control translates requests for CM from the CPU, increment unit, and I/O ports. The requests are translated for priority. The highest priority request is converted to a four-bit destination tag code and sent to SAS. This code, along with the two-bit I/O tag code, enters SAS when the address is accepted. These tags are kept with the addresses. If an address backup occurs, the tags are also backed up. The tags are used by destination control to ensure that CM data enters the selected destination register at the proper time.

The decoded destination tags are:

00	Not used
01	Read to IWS (fetch)
02	Exchange jump read
03	Return jump write
04	Not used
05	I/O port read
06	Exchange jump write
07	I/O port write
10	Not used
11	Increment read to X1
12	Increment read to X2
13	Increment read to X3
14	Increment read to X4
15	Increment read to X5
16	Increment write from X6
17	Increment write from X7

The decoded I/O tags are:

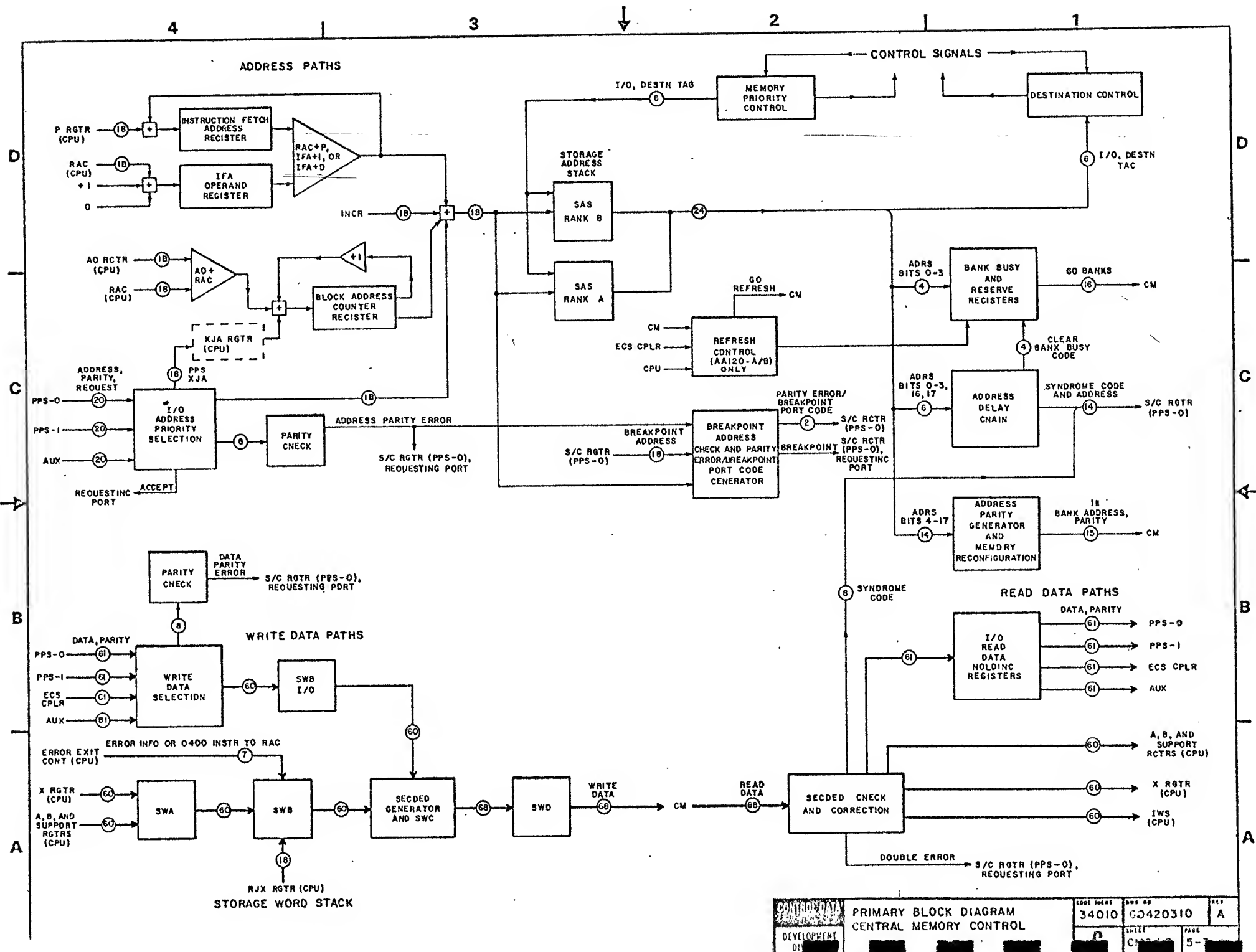
0	Auxiliary
1	PPS-1
2	PPS-0
3	ECS

The order of priority is:

1. ECS
2. CPU return jump or exchange jump
3. PPS-0
4. PPS-1
5. Auxiliary
6. CPU increment unit
7. CPU fetch

Requests from ECS and the CPU are mutually exclusive. ECS has top priority but is limited to a maximum rate of one request per 100 nanoseconds. ECS requests also have no conflicts in CM.

The CPU communicates with CMC in a different manner than the I/O ports. Requests due to instruction fetch and return/exchange jump have their addresses held in the CPU until the requests are honored by CMC. Requests for the increment unit must be ensured priority in CMC before the increment instruction issues because the increment unit has no request holding register.



## ADDRESS PATHS

The CMC address paths consist of address selection, SAS, breakpoint check, and address distribution.

### ADDRESS SELECTION

The SAS receives addresses from four sources: instruction fetch address (IFA) adder, block address counter (BAK) register, I/O address selection network (CMC 3.1), and increment unit (INCR 3.0). Addresses enter SAS under control of the memory priority translator (4LM7 module) shown on CMC 3.1.

#### INSTRUCTION FETCH ADDRESS

The IFA circuit (3IH7 and 3LL7 modules) contains two operand selection networks, two operand holding registers, and a two-stage adder. The adder output is gated into the SAS when a gate IFA to SAS signal has been generated by the memory priority translator. There are three possible outputs from the IFA adder. When either enter IFA-1 or -2 signal is generated by the coincidence test and fetch translator (CPU 3.3), the adder outputs  $P + RAC$ . When the advance IFA signal is generated by the memory priority translator, the adder output increases by one from the previous value. If neither of these signals is present, the adder output remains unchanged from the previous value. If both signals are present, the adder outputs  $P + RAC$ .

#### BLOCK ADDRESS COUNTER

The BAK circuit (4LB7 and 3HW7 modules) provides addresses for the SAS during exchange jump sequences and ECS block copies.

The BAK register and its associated static networks form the mechanism for counting twice through the 16 addresses required in an exchange sequence. The first eight words in the new exchange package are read from CM. The corresponding words in the previous package are then written into the same locations in CM. This process is repeated for the second eight words. The initial exchange sequence read and write address is XJA bits 0 through 17. This address enters the BAK register when the sample XJA signal is present. In the following 50-nanosecond clock period, this address enters the SAS. The address is incremented by one as each address enters the SAS. The incrementing is performed by the 4LB7 modules and is continued until all 16 addresses have twice entered the SAS. This process is controlled by the advance BAK and reload BAK signals. The reload BAK signal resets the BAK to the initial address after the first eight words have been read from CM. This allows the corresponding eight words in the previous package to be written into the same locations. This process is repeated for the second eight words.

The BAK register and its associated static networks form the mechanism for counting through all of the CM addresses in an ECS to CM or CM to ECS block copy. The initial CM address is  $A0 + RAC$  bits 0 through 17. This address is formed by the 4LB7 modules and enters the BAK register when the sample ECS address signal is present. In the following clock period, this address enters the SAS. The incrementing is performed by the 4LB7 modules and is continued until all CM addresses for the block copy have entered the SAS. This process is controlled by the advance BAK signal.

### STORAGE ADDRESS STACK

The SAS is a buffer for addresses arriving at CMC. Two 25-bit registers in the SAS are referred to as rank A and rank B. These registers hold an 18-bit CM address (3ST7 modules) and seven bits of control information (3SL7 module) shown on CMC 3.1. SAS allows CM requests to be accepted without the required bank being free.

If no CM bank conflicts occur (go or ECS active signal present), an address enters SAS rank A in any given 50-nanosecond clock period and leaves in the following clock period. Rank B is not used even if a new address enters rank A every 50-nanosecond clock period.

When a CM bank conflict occurs (go or ECS active signal not present), the address in rank A is held until the conflict is resolved. If another address arrives at SAS during this time, it enters rank B. When this backup condition occurs, no more addresses are issued to the SAS until the backup condition has been resolved. Once a backup condition has occurred, no new addresses are accepted by SAS until the addresses in both ranks have been sent to CM. Addresses leave SAS in the same order that they entered. (Rank A empties first; rank B empties last.)

#### BREAKPOINT CHECK

As addresses enter SAS, they are compared with the breakpoint address in the status and control register (PPS). If the addresses match, a breakpoint coincidence signal is sent to the 6SU7 module (CMC 3.4) for further testing.

#### ADDRESS DISTRIBUTION

The location of each word in CM is identified by an 18-bit address. Bits 0 through 3 are used to select one of 16 banks. Bit 3 also selects one of the two CSUs. Bits 4 through 15 select one of 32,768 words in a CSU quadrant. Bits 16 and 17 select one of four CSU quadrants.

#### BANK SELECTION

Bank selection is based upon the condition of address bits 0 through 3. Banks 0 through 7 are located in CSU-0. If address bit 3 is clear, one of banks 0 through 7 is selected. If address bit 3 is set, one of banks 10 through 17 in CSU-1 is selected.

The 3SN7 modules contain 16 bank busy registers which correspond to the 16 banks in CM. These registers keep track of which banks are currently active with a memory reference. Address bits 0 through 3 determine which bank busy register is set and which bank receives a go bank signal to start a memory reference. The 3SM7 module ensures that the requested bank is free. If the bank is already busy, the set bank busy and go bank signals are not generated until the bank is free. Bank busy registers are cleared 350 nanoseconds after they are set by the seven-rank delay chain (4KK7 and 6XD7 modules) and the full signal from destination control (CMC 3.3).

The 3SN7 modules also contain 16 bank reserve registers which correspond to the 16 banks in CM. These registers ensure that a bank required for an ECS transfer is free. This allows the block copy to proceed at a rate of one word per 100 nanoseconds. The bank reserve registers are set under control of the bank reserve generator in the 3SM7 module. Bank reserve registers are set for bank N+1, N+2, N+3, and N+4 when bank N busy register is set by an ECS transfer. Bank reserve registers are cleared when the corresponding bank busy register is set or when the clear reservation signal from CMC 3.5 is present. This indicates a record gap in the ECS transfer. For non-ECS references, reserved banks appear busy and cannot be accessed until they are not busy and not reserved.

#### CSU ADDRESS

The SAS transmits address bits 4 through 17 (via 4SX7 module) to both CSUs during the same 50-nanosecond clock period that the go bank signal is transmitted. Address parity is generated on the 4SX7 module and delayed 50 nanoseconds in the 3SW7 module. Also, address bits 15 through 17 pass through the memory reconfiguration translator on the 4SX7 module.

## CM ERROR ADDRESS

The 10-rank address delay chain receives address bits 0 through 3 (bank select) from SAS and address bits 16 and 17 (quadrant select) from the memory reconfiguration translator. These bits are delayed and sent to the status and control register along with syndrome code bits 0 through 7 from the SECDED check network (CMC 3.2). This information can then be interpreted to isolate the CM failure.

## MEMORY RECONFIGURATION

Address bits 15 through 17 can be manipulated so that a failing quadrant of CM is bypassed to provide a continuous block of usable CM. The memory reconfiguration circuit consists of four two-position switches located on chassis 5 and a translator on the 4SX7 module. Only one reconfiguration is available for a normally configured CM. A reconfigured CM exhibits the same sequential addressing characteristics as a normally configured CM of the same size. Reconfigurations available are:

<u>Normal Memory Size</u>	<u>Reconfigured Memory Size</u>
65K	No reconfiguration
98K	65K (only if 32K portion fails)
131K	65K
196K	131K
262K	196K

## WRITE DATA PATHS AND CONTROL

CMC transmits a 60-bit data word (plus 8-bit SECDED code) to both CSUs of CM. The data is transmitted during the 50-nanosecond clock period following the go bank signal (CMC 3.0). Two write data paths exist, one path for I/O write data from PPS-0, PPS-1, ECS, and auxiliary ports, and the other for write data from the CPU. The two paths are merged in the SWS before the data is sent to CM.

### I/O WRITE DATA PATH

The I/O write data path consists of four write data holding registers, a selection network, parity checking network, and SWS ranks B through D (SWB through SWD). I/O write data is merged with CPU write data at the input to SWC.

The 3S07 modules contain a separate 60-bit (plus parity bit) write data holding register for each I/O port. Each register functions independently. Data enters the registers during any clock period that the corresponding hold signal is not present. Data is held in the registers during the time that the corresponding hold signal is present.

One of the registers is selected for use by I/O tag bits 0 and 1. The selected data is gated to the SWB I/O register (4HM7 modules) when the shift SWB signal is present. The SWB I/O register is used to hold I/O data during an SAS backup condition (CMC 3.0).

The selected data is partially parity checked by the 3S07 modules. Twelve 3S07 modules generate eight parity check bits. These bits are sent to a 6SQ7 module (CMC 3.4) where the parity check is completed. An I/O data parity error signal is then sent to an error reporting network (CMC 3.4). No action on errors is taken until the requested address is accepted by the SAS.

### CPU WRITE DATA PATH

The CPU write data path consists of SWS ranks A through D (SWA through SWD). The SWS is a buffer area for 60-bit words which are to be written into CM. Ranks A and B are contained in 4HM7 modules, rank C is contained in 3XH7 modules, and rank D is contained in 3SW7 modules.

#### SWA

SWA is the first rank of the SWS. This register is normally cleared and entered with the content of XI during each 25-nanosecond clock period. This data is shifted through the four ranks of SWS and written into CM only when an increment instruction is issued which requires a CM reference. In all other cases, the data is shifted through the ranks and discarded. This normal mode of operation is altered during an exchange sequence.

During the first half of an exchange sequence, the memory priority translator generates a switch SWA for exchange signal. This signal blocks the XI register data and allows the first eight 60-bit words of the exchange package (A, B, and support register data) to be shifted through the SWS into CM. The last eight words of the exchange package consist of X register data which is shifted through the SWS in the normal mode of operation.

#### SWB

SWB is the second rank of the SWS. It consists of two 60-bit registers which are normally cleared and entered with write data every 50-nanosecond clock period. Only one of the two SWB registers can contain valid write data at a given time. The SWB I/O register was described earlier. The SWB register is normally cleared and entered with SWA during each clock period. This normal mode of operation is altered in the event of a CM backup or during a return jump sequence.

The decoded I/O tags are:

- |   |           |
|---|-----------|
| 0 | Auxiliary |
| 1 | PPS-1     |
| 2 | PPS-0     |
| 3 | ECS       |

The order of priority is:

1. ECS
2. CPU return jump or exchange jump
3. PPS-0
4. PPS-1
5. Auxiliary
6. CPU increment unit
7. CPU fetch

Requests from ECS, PPS-0, PPS-1, and auxiliary ports are processed by the I/O priority network (CMC 3.5). Requests from ECS and the CPU are mutually exclusive. ECS has top priority but is limited to a maximum rate of one request per 100 nanoseconds. Also, ECS requests have no conflicts in CM.

The CPU communicates with CMC in a different manner than the I/O ports. Requests because of instruction fetch and return/exchange jump have their addresses held in the CPU until the requests are accepted by CMC. Requests for the increment unit are ensured priority in CMC before the increment instruction issues because the increment unit has no address, data, and request holding registers.

#### I/O ADDRESS

The 3S07 modules contain a separate 18-bit (plus parity bit) address holding register for each I/O port. Each register functions independently. The PPS-0 and PPS-1 registers are used for CM references and exchange jump requests. The auxiliary register is used for CM references and starting CM addresses for ECS transfers. The ECS register is unused since starting CM addresses for ECS transfers are formed in the BAK register (CMC 3.0). Addresses enter the registers coincident with their requests (CMC 3.5) when the corresponding hold signal is not present. Addresses are held in the registers during the time that the corresponding hold signal is present.

One of the registers is selected for use by I/O tag bits 0 and 1. The selected address is sent to the SAS (CMC 3.0) and XJA register (CPU 3.7).

The selected address is partially parity checked by the 3S07 modules. Ten 3S07 modules generate eight parity check bits. These bits are sent to a 6SQ7 module (CMC 3.4) where the parity check is completed. An I/O address parity error signal is then sent back to the SAS (3SL7 module) and to an error reporting network (CMC 3.4). No action on errors is taken until the request is accepted by the SAS.

## READ DATA PATH

The read data path consists of the read data holding register (3XH7 modules), the SECDED network (3XH7 and 4XI7 modules), and the I/O read data holding registers (3SS7 modules).

### READ DATA HOLDING REGISTER

This register receives a 60-bit (plus 8-bit SECDED code) read data word from CM every 50-nanosecond clock period. Only one of the two CSUs of CM sends valid data during any given clock period. The other CSU sends a 68-bit word of all ones. If address bit 3 is set, CSU-1 data enters the holding register. If clear, CSU-1 data enters. Read data enters the holding register ten 50-nanosecond clock periods after a go bank signal (CMC 3.0) is sent to CM.

### SECDED NETWORK

The read data path provides SECDED or odd parity checking on data read from CM. A switch located on chassis 5 selects the SECDED or parity mode of operation. SECDED is the normal mode and is depicted by the simplified block diagram shown in Figure 5-4. The SECDED network is also used to generate and check parity.

### SECDED CODE GENERATOR (shown on CMC 3.1)

An 8-bit SECDED code is generated when a 60-bit write data word is held in SWC. The SECDED code and write data are shifted through SWD and are sent to CM.

The SECDED code is generated (3XH7 modules) according to the matrix shown in Figure 5-5. SECDED code bits 0 through 7 are represented by X's in the corresponding columns. Write data bits 0 through 63 (bits 60 through 63 constant zeros) which are to be parity checked are represented by X's in the corresponding columns.

Each SECDED code bit is generated by performing an odd parity generate on the 26 bits of write data in the corresponding row. If an odd number of one bits is present, the corresponding SECDED code bit is a zero. If an even number of one bits is present, the SECDED code bit is a one.

### SYNDROME CODE GENERATOR

An 8-bit syndrome code is generated when a 60-bit word (plus 8-bit SECDED code) is read from CM into the read data holding register. The SECDED logic algorithm used in design requires 64 data bits. Therefore, the upper four bits (60 through 63) are forced to constant zeros. The entire 72 bits of read data and SECDED code are used to form the syndrome code. This circuit also generates a transmission parity bit (P60 generate signal) and sends it to the requesting port. If a single data bit error occurs, the transmission parity bit is inverted to compensate for the single-error correction.

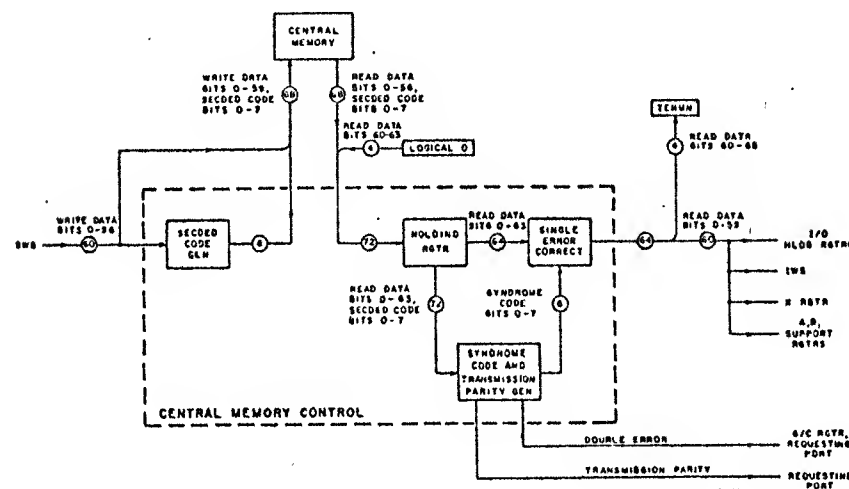


Figure 5-4. SECDED Network Block Diagram (SECDED Mode)

The syndrome code is used to detect and correct errors. Syndrome codes are listed in Table 5-5. All possible syndrome codes from 000 through 377 (octal) are listed and described in this table. The bit column indicates which bit failed or references a note at the bottom of the table. This note explains all codes which do not indicate a single-bit error.

The eight syndrome code bits along with six address bits associated with the memory reference are sent to the status and control register. This information can then be interpreted to allow determination of failing CSU, memory bank, memory quadrant, and in the case of single correctable errors, the failing bit. This information makes it possible for maintenance personnel to isolate the failure to a module level.

When a single data bit fails, a syndrome code containing three or five one bits is generated. The single-error correction network (4X17 modules) decodes the syndrome. If the syndrome code corresponds to the code for any bit position shown in the matrix (Figure 5-5), the bit is assumed to be in error. This bit is inverted by the single-error correction network. Therefore, operation continues unimpeded despite a single-bit CM failure. When the syndrome code does not correspond to any of the codes in the matrix (Figure 5-5), the read data passes through the single-error correction network unchanged.

The code in the matrix can be ascertained by reading the X's in the column for a particular bit.

[illegible]

Figure 5-5. SECDED Matrix

If two data bits fail, a syndrome code containing an even number of one bits is generated. No correction is made and the double-error bit sets in the status and control register.

If a multiple bit failure occurs, the SECDED network treats a resulting syndrome code containing an even number of bits as a double error. A resulting syndrome code with an odd number of bits is treated as a single error. However, some combinations of multiple-bit failures may result in a legitimate single-error syndrome code. This results in complementing a bit that may or may not have been correct.

The operation of the SECEDED network in parity mode is depicted by the simplified block diagram shown in Figure 5-6. A single odd parity bit is generated when a 60-bit write data word is held in SWC (CMC 3.1). This parity bit is in the position normally occupied by SECEDED code bit 0. The parity bit and write data are shifted through SWD and are sent to CM.

When the 60-bit (plus parity bit) word is read from CM, the word passes unchanged through the single-error correction network. Also, a transmission parity bit is generated (P 60 generate signal). If P 60 generate and the parity bit received from CM are unequal, a parity error is reported to the exit condition register (CPU 3, 10) and the transmission parity bit is complemented before being sent to the requesting port.

CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT
000	⑤	040	85 ①	100	66 ①	140	②	200	67 ①	240	②	300	②	340	50		
001	80 ①	041	②	101	②	141	53	201	②	241	37	301	56	341	②		
002	61 ①	042	②	102	②	142	54	202	②	242	59	302	①	342	②		
003	⑤	043	0	103	1	143	②	203	2	243	②	303	②	343	③		
004	62 ①	044	②	104	②	144	40	204	②	244	①	304	①	344	②		
005	②	045	23	105	3	145	②	205	5	245	②	305	②	345	①		
006	②	046	22	106	6	146	②	206	6	246	②	306	②	346	③		
007	10	047	②	107	②	147	②	207	②	247	44	307	④	347	②		
010	63 ①	050	②	110	②	150	41	210	②	250	43	310	48	350	②		
011	②	051	47	111	7	151	②	211	8	251	②	311	②	351	28		
012	②	052	27	112	31	152	②	212	11	252	②	312	②	352	①		
013	13	053	②	113	②	153	③	213	②	253	①	313	③	353	②		
014	②	054	29	114	30	154	②	214	18	254	②	314	②	354	①		
015	17	055	②	115	②	155	③	215	②	255	①	315	③	355	②		
016	16	056	②	116	②	156	③	216	②	256	③	316	③	356	②		
017	②	057	③	117	52	157	②	217	③	257	②	317	③	357	③		
020	64 ①	060	②	120	②	160	42	220	②	260	45	320	49	360	②		
021	②	061	46	121	51	161	②	221	56	261	②	321	②	361	③		
022	②	062	32	122	55	162	②	222	15	262	②	322	②	362	①		
023	14	063	②	123	②	163	③	223	③	263	③	323	36	363	②		
024	②	064	33	124	35	164	②	224	39	264	②	324	②	364	20		
025	19	065	②	125	②	165	③	225	②	265	③	325	③	365	②		
026	21	066	②	126	②	166	③	226	②	266	③	326	③	366	②		
027	②	067	①	127	③	167	②	227	③	267	②	327	②	367	③		
030	②	070	34	130	37	170	②	230	36	270	②	330	②	370	③		
031	24	071	②	131	②	171	③	231	②	271	③	331	③	371	②		
032	25	072	②	132	②	172	12	232	②	272	③	332	③	372	②		
033	②	073	③	133	③	173	②	233	③	273	③	333	②	373	③		
034	26	074	②	134	②	174	③	234	②	274	③	334	③	374	②		
035	②	075	4	135	①	175	②	235	③	275	②						

- ① Syndrome code bit failed (single code bit set).
- ② Double error or multiple error (even number of code bits set).
- ③ Multiple error reported as single error (five or seven code bits set).
- ④ Not used because of 64-bit algorithm.
- ⑤ No error was detected.

## READ DATA DISTRIBUTION

Read data from the SECDED network is sent to the I/O read data holding registers, the IWS, the X registers, and the A, B, and support registers. Data to the X registers is gated by the read to X signal from destination control (CMC 3.3). If read to X is not present, all zeros are sent to the X registers. Data to the A, B, and support registers is gated by the gate exchange data signal (CPU 3.7) which is present during the first eight words of an exchange read sequence. If this signal is not present, all ones are sent to the A, B, and support registers.

Each I/O port (PPS-0, PPS-I, ECS, and auxiliary) has a separate 60-bit read data holding register (3SS7 modules). Data enters a register under control of the I/O read signal and I/O tag bits 0 and 1. Data remains in a register until new data is entered in the same register. The output of each register is transmitted to its respective port. The registers also hold and transmit the transmission parity and double-error signals to the ports.

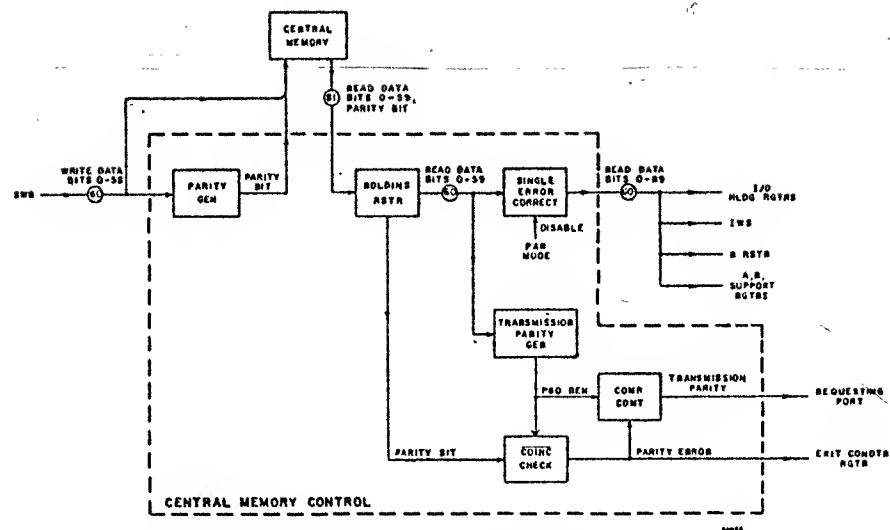


Figure 5-6. SECDED Network Block Diagram (Parity Mode)

## DESTINATION CONTROL

Destination control consists of two control modules (4SJ7 and 6SV7 modules) and a 10-rank delay chain. The four-bit destination tag and two-bit I/O tags in SAS (CMC 3.1) are decoded and delayed to enable various control signals required for a CM reference. Error monitoring is also performed by destination control.

### CM READ CONTROL

The control signals generated for a CM read reference depend upon the four-bit destination tag code. The decoded destination tags for a read operation are:

- 01 Read to IWS (fetch)
- 02 Exchange jump read
- 05 I/O port read
- 11 Increment read to X1
- 12 Increment read to X2
- 13 Increment read to X3
- 14 Increment read to X4
- 15 Increment read to X5

### CM WRITE CONTROL

The control signals generated for a CM write reference also depend upon the four-bit destination tag code. The decoded destination tags for a write operation are:

- 03 Return jump write
- 06 Exchange jump write
- 07 I/O port write
- 16 Increment write from X6
- 17 Increment write from X7

### CSU ADDRESS PARITY ERROR

The 4SJ7 module receives an address parity error signal from a CSU if an error occurs when the address is sent from the SAS (CMC 3.0) to CM. This signal is retransmitted to the status and control register three 50-nanosecond clock periods after the go bank signal (CMC 3.0) was transmitted to CM. A CSU address parity error causes a write operation to be blocked (CM unchanged) and forces read data from CM to all ones. An error signal is also sent to the requesting I/O port or to the CPU exit condition register (CPU 3.10).

### INCREMENT RANGE ERROR

If an increment address range error occurs and an increment address is in SAS-A or -B, the corresponding increment error in SAS flag sets. These flags determine whether or not increment data is enabled to an X register. If either flag is set, the corresponding read data is discarded and replaced with all zeros.

### SECEDED ERROR

A SECEDED error signal is sent to the status and control register if the syndrome code is not zero when CM is read. The SECEDED error signal (CMC 3.2) input to the 6SV7 module indicates that one or more syndrome code bits is set.

### DOUBLE ERROR

A double-error signal is sent to the requesting I/O port and the status and control register when CM is read if the syndrome code is not zero and an even number of bits is set. The parity check network (6SV7 module) tests for an odd or even number of bits set in the syndrome code.

#### PARITY OR DOUBLE ERROR

This signal is sent to the CPU exit condition register (CPU 3.10) if a double error occurs when CM is read. If the SECDED network is set to parity mode, this signal indicates that a parity error occurred when CM was read.

#### TRANSMISSION PARITY

Read data sent to an I/O port is accompanied by an odd parity bit. This bit is generated when data is read from CM. If a CM parity error (parity mode) occurs, the transmission parity bit is complemented. This allows the CM error to be reported as a transmission error at the requesting port. If a single data bit error (SECDED mode) occurs, the transmission parity bit is complemented. This compensates for the corrected bit which was complemented by the SECDED network.

## I/O ACCEPT, BREAKPOINT, PARITY CHECK

### FORCE ZERO SECDED CODE AND PARITY

The fanout for this signal is located in the 4LU7 and 3XE7 modules. This signal is received from the status and control register. It forces to zero all data parity and SECDED code bits sent from CMC.

### PARITY CHECK

The 6SQ7 modules perform the second half of the I/O write data and address parity check. The first half is performed in 3S07 modules (CMC 3.1) on data and addresses from PPS-0, PPS-1, ECS, and auxiliary ports. (The ECS address port is unused.)

A CMC parity error signal is sent from the 6SU7 module to the status and control register if an error is sensed on the address or write data in the I/O input holding registers (CMC 3.1). The CMC parity error signal is transmitted when the accept signal is sent to the requesting I/O port.

A CMC address/data parity error signal is sent from the 6SU7 module to the status and control register when a parity error is sensed in the I/O address holding register (CMC 3.1). This signal is transmitted when the accept signal is sent to the requesting I/O port.

### CMC INPUT ERROR

A CMC input error signal is sent from the 6SU7 module to the requesting I/O port. This signal is transmitted when the accept signal is sent to the requesting I/O port.

A CMC input error signal is sent to the ECS coupler when the write data at the ECS input holding register (CMC 3.1) has incorrect parity. The register is checked for correct parity on ECS write of CM only.

A CMC input error signal is sent to the PPS when the write data or the address in the PPS input holding register (CMC 3.1) has incorrect transmission parity. The input data register parity is checked on write operations only. If an input address error occurs on a memory reference, the write operation is blocked (CM data is unchanged) and the data for the read operation is replaced with a word of all ones. If an input error occurs on an exchange request, an accept signal is sent but the exchange instruction does not execute.

A CMC input error signal is sent to the auxiliary port when the write data or the address in the auxiliary input holding register (CMC 3.1) has incorrect transmission parity. The input address register parity is checked on both read and write operations. If an input error is caused by an address parity error, the write operation is blocked (CM data is unchanged) and the data for the read operation is replaced by a word of all ones.

### I/O ACCEPT

An accept signal (6SU7 module) is sent to the requesting I/O port when the current request for CM access has been processed to the point where another request can be received at the 6SP7 module (CMC 3.5).

### ECS ACCEPT

In ECS mode, an accept signal is sent to the ECS coupler when the 3SM7 module (CMC 3.0) sends a go CSU signal to CM.

### AUXILIARY ACCEPT

In ECS mode, an accept signal is sent to the auxiliary port when the 3SM7 module (CMC 3.0) sends a go CSU signal to CM. If CMC is not in ECS mode, an accept signal is sent to the auxiliary port when the address for this CM reference enters the SAS.

#### PPS-0, -1 ACCEPT

In ECS mode, an accept signal is sent to the selected PPS port when the 3SM7 module (CMC 3.0) sends a go CSU signal to CM. If CMC is not in ECS mode, an accept signal is sent to the selected PPS port when the address for this CM reference enters the SAS.

The same accept signal is used for CM references and exchange jump requests. However, the two requests cannot be handled at the same time. The accept signal for an exchange jump request is transmitted 25 nanoseconds after the exchange sequence flag (CPU 3.7) sets.

#### BREAKPOINT

The 6SU7 module monitors the breakpoint coincidence signals from the 3ST7 modules (CMC 3.0). Breakpoint coincidence occurs when the breakpoint address in the status and control register matches an address at the input to the SAS (CMC 3.0). When the addresses match, all three breakpoint coincidence signals are generated. A breakpoint condition code from the status and control register determines whether or not a breakpoint signal is sent to the requesting I/O port or to error exit control (CPU 3.10). The status and control register also receives the breakpoint signal, a breakpoint function code, and a breakpoint and parity error port code.

#### BREAKPOINT CONDITION CODE

Breakpoint condition code bits 18 through 21 from the status and control register select what type of memory reference will generate a breakpoint signal when a breakpoint condition occurs. The selections are:

<u>21</u>	<u>20</u>	<u>19</u>	<u>18</u>	<u>Breakpoint On</u>
0	0	X	X	None
0	1	X	X	PPS
1	0	X	X	CPU and auxiliary
1	1	X	X	PPS, CPU, and auxiliary

<u>21</u>	<u>20</u>	<u>19</u>	<u>18</u>	<u>Breakpoint On</u>
X	X	0	0	Any read
X	X	0	1	Any write
X	X	1	0	Fetch
X	X	1	1	Any reference

#### BREAKPOINT FUNCTION CODE

A two-bit breakpoint function code is sent to the status and control register to indicate what type of memory reference caused a breakpoint condition. The indications are:

<u>1</u>	<u>0</u>	<u>Reference</u>
0	0	Read
0	1	Write
1	0	Fetch
1	1	Not used

#### BREAKPOINT AND PARITY ERROR PORT CODE

A two-bit code is sent to the status and control register to indicate which requesting port caused a breakpoint condition. The same two-bit code is also sent to indicate which port caused a parity error. The indications are:

<u>1</u>	<u>0</u>	<u>Requesting Port</u>
0	0	Not used
0	1	CPU
1	0	PPS-1
1	1	PPS-0

## I/O EXCHANGE, I/O PRIORITY, AND REFRESH CONTROL

### I/O EXCHANGE

The 6SQ7 module receives and processes exchange requests for CM access from PPS-0 and PPS-1. Each PPS has four exchange registers. The registers are set by the exchange request, 261 instruction, 262 instruction, and exchange CPU define signals from the PPS and are all cleared when the exchange resume signal is received from the CPU. This indicates that the CPU has accepted the exchange request. The signals received from the PPS are:

Exchange request	This signal informs CMC that the PPS has executed an exchange jump instruction.
261 instruction	This signal informs CMC that the exchange request was caused by a 261 instruction (monitor exchange jump-A). It arrives at the same time as the exchange request signal.
262 instruction	This signal informs CMC that the exchange request was caused by a 262 instruction (monitor exchange jump-MA). It arrives at the same time as the exchange request signal.
Exchange CPU define	This signal informs CMC that the exchange request is for CPU-1. Since CPU-1 does not exist at this time, the signal causes CPU-0 to exchange.

#### NOTE

Portions of the 6SQ7 module have been provided to allow future enhancement to convert the auxiliary port to a CPU-1 port.

### I/O PRIORITY

The 6SP7 module receives and processes requests for CM access from ECS, PPS-0, PPS-1, and auxiliary ports. Requests are handled by a priority network which ensures that each requesting port has been accepted before a higher priority requestor regains priority. The priority network has a 50-nanosecond dead space when it resets, allowing the CPU to access CM periodically when I/O activity is heavy. The order of I/O priority is:

1. ECS
2. PPS-0
3. PPS-1
4. Auxiliary

### REQUEST/ACCEPT

The I/O request signals are held in registers at the input to the I/O priority translator. The data, address, and write signals are held in the input holding registers (CMC 3.1) while the corresponding request register is set. A request register is cleared when the corresponding accept signal (CMC 3.4) is sent to the requesting port.

Requests from PPS-0, PPS-1, and auxiliary ports inform CMC that an address has been sent (CMC 3.1) to initiate a CM read or write reference.

### ECS MODE

CMC provides a dual mode of operation. When an ECS transfer is in process, CMC is in ECS mode. In ECS mode, the CPU cannot reference CM, banks are reserved for ECS, and address backup and conflicts in SAS cannot occur.

If an address for an I/O reference enters SAS and causes a conflict, the address is written over by an address for an ECS reference 50 nanoseconds later. Addresses that conflict in this manner continue to be reentered in SAS until the required bank is not busy and not reserved. When CMC is not in ECS mode, CMC cannot respond to its ECS port, and the CPU and SAS function normally. The ECS coupler controls block transfer by sending control signals to the 6SP7 module. These signals are:

ECS request	This signal switches CMC into ECS mode. A go ECS signal is returned to the ECS coupler when the ECS request register sets.
ECS CPU define	This signal indicates which CPU has initiated the transfer. If set, CPU-0 provides the starting CM address for the BAK register (CMC 3.0). If clear, CPU-1 (auxiliary port) provides the address.
ECS bank initiate	This signal is the request for each ECS reference of a CM bank. ECS mode must be selected before CMC can sense this signal.
ECS continue request	This signal reserves CM banks 400 nanoseconds before they are required for a reference. Therefore, it must arrive 400 nanoseconds before each ECS bank initiate signal. ECS mode must be selected before CMC can sense this signal.
ECS end of transfer	This signal terminates all ECS activity in CMC by clearing the ECS request register. CMC is then free to accept and process other requests.

#### REFRESH CONTROL

The 5SR7 module controls refresh references in the 16 CM banks.<sup>†</sup> Refresh is in effect a CM read operation. Therefore, the banks to be refreshed must be free. All banks are refreshed every 25.6 microseconds. Refresh margins may be applied under control of the status and control register to change the rate to 32.0 (slow) or 19.4 (fast) microseconds.

<sup>†</sup> 5SR7 module is not present in AA120-C.

60420310 A

#### GO REFRESH

A go refresh signal is sent (through a fanout shown on CMC 3.0) to all 16 banks in CM. Each signal causes a refresh reference to be started in the corresponding bank. The signals are sent in two groups of eight so that all eight banks of a CSU start a refresh reference at the same time. All 16 banks can start at the same time (normal mode) or at two different times (ECS mode). Go refresh is never sent within 400 nanoseconds of a go bank signal (CMC 3.0) to the same CSU.

#### NORMAL MODE

When ECS transfers are not present, the refresh period counter generates a refresh request every 25.6 microseconds. This sets the refresh active register if an exchange sequence is not in process (XSF not set). If XSF (CPU 3.7) sets, the refresh active register waits until XSF clears before it sets. After the refresh active register sets, all CM and exchange requests are blocked and the refresh delay counter starts a 400-nanosecond delay count. When the delay is completed, all CM banks are free and a register sets for each CSU chassis. These registers each send a go refresh signal to eight banks. The refresh active register is cleared 400 nanoseconds after it sets by the refresh delay counter. This restores CM to normal operation.

#### ECS MODE

When an ECS transfer is present, refresh references are phased between the two CSU chassis without affecting the ECS transfer rate. While ECS is referencing bank 4 (CSU-0), CSU-1 is refreshed. While ECS is referencing bank 14 (CSU-1), CSU-0 is refreshed. I/O requests for CM are blocked at this time. If a record gap of 800 nanoseconds in the ECS transfer occurs, the refresh reference is delayed until address bits 0 through 3 indicate that bank 4 or 14 has received a go bank signal (CMC 3.0). If the record gap is longer than 800 nanoseconds, the refresh reference continues to completion without waiting for a go bank signal.

# FETCH SEQUENCE

Time	Control	Address	Data
	<ol style="list-style-type: none"> <li>1. Set F1F (6HK7).</li> <li>2. Destination tag 01 and go address (4LM7).</li> <li>3. Gate IFA to SAS (4LM7).</li> <li>4. Advance IFA (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>1. Fetch address formed in IFA adder (3HI7, 3LL7).</li> </ol>	
T0	<ol style="list-style-type: none"> <li>5. Destination tag 01 and go address (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>2. Address in SAS (3ST7).</li> <li>3. Address flag from SAS tags and control (3SL7).</li> <li>4. Address bits 4-17 in memory reconfiguration and parity generator (4SX7).</li> <li>5. Address bits 0-3 in bank select (3SM7).</li> <li>6. Address bits 0-3 in refresh control (5SR7).†</li> <li>7. Transmit address bits 4-17 (4SX7).</li> <li>8. Go CSU (3SM7).</li> <li>9. Generate set bank busy (3SM7).</li> </ol>	
T1	<ol style="list-style-type: none"> <li>6. Destination tag 01 in delay rank 1 (4SJ7).</li> </ol>	<ol style="list-style-type: none"> <li>10. Set bank busy (3SN7).</li> <li>11. Transmit go bank (3SN7) and address parity bit (3SW7).</li> <li>12. Address bits 16, 17 in delay rank 1 (5XD7).</li> <li>13. Address bits 0-3 in delay rank 1 (4KK7).</li> </ol>	
T2	<ol style="list-style-type: none"> <li>7. Destination tag 01 in delay rank 2 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>14. Address bits 16, 17 in delay rank 2 (5XD7).</li> <li>15. Address bits 0-3 in delay rank 2 (5XD7).</li> </ol>	
T3	<ol style="list-style-type: none"> <li>8. Destination tag 01 in delay rank 3 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>16. Address bits 16, 17 in delay rank 3 (5XD7).</li> <li>17. Address bits 0-3 in delay rank 3 (5XD7).</li> </ol>	
T4	<ol style="list-style-type: none"> <li>9. CSU address parity error in destination control-1 (delay rank 4) (4SJ7).</li> <li>10. CSU address parity error sent to s/e register.</li> <li>11. Destination tag 01 in delay rank 4 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>18. Address bits 16, 17 in delay rank 4 (5XD7).</li> <li>19. Address bits 0-3 in delay rank 4 (5XD7).</li> </ol>	
T5	<ol style="list-style-type: none"> <li>12. Destination tag 01 in delay rank 5 (5XD7).</li> <li>13. CSU address parity error in delay rank 5 (4KK7).</li> <li>14. CSU address parity error sent to destination control-1 (4SU7).</li> <li>15. CMC input error sent to error sense circuit (4SY7).</li> </ol>	<ol style="list-style-type: none"> <li>20. Address bits 16, 17 in delay rank 5 (5XD7).</li> <li>21. Address bits 0-3 in delay rank 5 (5XD7).</li> </ol>	
T6	<ol style="list-style-type: none"> <li>16. Destination tag 01 in delay rank 6 (4SU7).</li> </ol>	<ol style="list-style-type: none"> <li>22. Address bits 16, 17 in delay rank 6 (5XD7).</li> <li>23. Address bits 0-3 in delay rank 6 (5XD7).</li> </ol>	
T7	<ol style="list-style-type: none"> <li>17. Destination tag 01 in delay rank 7 (4SJ7).</li> <li>18. Destination tag 01 in delay rank 7, full (enable to clear bank busy) (4SJ7).</li> <li>19. Full fanout (4LU7).</li> </ol>	<ol style="list-style-type: none"> <li>24. Address bits 16, 17 in delay rank 7 (5XD7).</li> <li>25. Address bits 0-3 in delay rank 7 (5XD7).</li> <li>26. Address bits 0-3 fanout (4LU7).</li> </ol>	
T8	<ol style="list-style-type: none"> <li>20. Destination tag 01 in delay rank 8 (4KK7).</li> </ol>	<ol style="list-style-type: none"> <li>27. Address bits 16, 17 in delay rank 8 (5XD7).</li> <li>28. Address bits 0-3 in delay rank 8 (4KK7).</li> <li>29. Address bits 0-3 clear bank busy (3SN7).</li> </ol>	

† 5SR7 module is not present in AA120-C.

# FETCH SEQUENCE (Cont'd)

ime	Control	Address	Data
T9	21. Destination tag 01 in delay rank 9 (4KK7).	30. Address bits 16, 17 in delay rank 9 (5XD7). 31. Address bits 0-3 in delay rank 9 (5XD7). 32. Address bit 3 in delay rank 9 (4KK7). 33. Address bit 3 (select CSU-0) fanout (3XE7, 4LU7).	
T10	22. Destination tag 01 in delay rank 9 sent to destination control-2, shift stack signal generated (6SV7). 23. Destination tag 01 in delay rank 10, double error, parity or double error (6SV7).	34. Address bits 16, 17 delay rank 10 (5XD7). 35. Address bits 0-3 in delay rank 10 (5XD7).	1. CSU read data sent to holding register, and syndrome generator (3XH7). 2. SECDED correction and data distribution (4XI7). 3. Read data bits 0-59 sent to IWS (4HIB7). 4. Syndrome code bits 0-7 and SECDED error signal sent to destination control-2 (6SV7).
T11	24. Double error-1 and SECDED error-1 signals sent to s/c register (6SV7).	36. Address information to s/c register (3SW7).	

# RETURN JUMP SEQUENCE

ime	Control	Address	Data
	1. Set SXF (6HK7). 2. Destination tag 03 and go address (4LM7). 3. Gate IFA to SAS (4LM7). 4. Switch SWB for RJX (4LM7). 5. Shift SWA, SWB, SWC (4LM7).	1. P, RAC bits 0-17 enter IFA adder (3HI7, 3LL7)	1. Xi register data to SWA (4HM7).
T0	6. Destination tag 03 and go address (3SL7).	2. Address in SAS (3ST7). 3. Address flag from SAS tags and control (3SL7). 4. Address bits 4-17 in memory and reconfiguration and parity generator (4SX7). 5. Address bits 0-3 in bank select (3SM7). 6. Address bits 0-3 in refresh control (5SR7). <sup>†</sup> 7. Transmit address bits 4-17 (4SX7). 8. Go CSU (3SM7). 9. Generate set bank busy (3SM7).	2. RJX bits 30-47 and SWB bit 56 (set) to SWB (4HIM7). 3. Return jump write data to SWC and SECDED code generator (3XH7).
T1	7. Destination tag 03 in delay rank 1 (4SJ7). 8. Transmit write signal (4SJ7).	10. Set bank busy (3SN7). 11. Transmit go bank (3SN7) and address parity bit (3SW7). 12. Address bits 16, 17 in delay rank 1 (5XD7). 13. Address bits 0-3 in delay rank 1 (4KK7).	4. Return jump write data and SECDED code to SWD (3SW7). 5. Transmit return jump write data and SECDED code (3SW7).
T2	9. Destination tag 03 in delay rank 2 (5XD7).	14. Address bits 16, 17 in delay rank 2 (5XD7). 15. Address bits 0-3 in delay rank 2 (5XD7).	
T3	10. Destination tag 03 in delay rank 3 (5XD7).	16. Address bits 16, 17 in delay rank 3 (5XD7). 17. Address bits 0-3 in delay rank 3 (5XD7).	
T4	11. CSU address parity error in destination control-1 (delay rank 4) (4SJ7). 12. CSU address parity error sent to s/c register. 13. Destination tag 03 in delay rank 4 (5XD7).	18. Address bits 16, 17 in delay rank 4 (5XD7). 19. Address bits 0-3 in delay rank 4 (5XD7).	

<sup>†</sup>5SR7 module is not present in AA120-C.

RETURN JUMP SEQUENCE (Cont'd)

Time	Control	Address	Data
T5	14. Destination tag 03 in delay rank 5 (5XD7). 15. CSU address parity error in delay rank 5 (4KK7). 16. CSU address parity error sent to destination control-1 (4SJ7). 17. CMC input error sent to error sense circuit (4SY7).	20. Address bits 16, 17 in delay rank 5 (5XD7). 21. Address bits 0-3 in delay rank 5 (5XD7).	
T6	18. Destination tag 03 in delay rank 6 (4SJ7).	22. Address bits 16, 17 in delay rank 6 (5XD7). 23. Address bits 0-3 in delay rank 6 (5XD7).	
T7	19. Destination tag 03 in delay rank 7 (4SJ7). 20. Destination tag 03 in delay rank 7, full (enable to clear bank busy) (4SJ7). 21. Full fanout (4LU7).	24. Address bits 16, 17 in delay rank 7 (5XD7). 25. Address bits 0-3 in delay rank 7 (5XD7). 26. Address bits 0-3 fanout (4LU7).	
T8	22. Destination tag 03 in delay rank 8 (4KK7).	27. Address bits 16, 17 in delay rank 8 (5XD7). 28. Address bits 0-3 in delay rank 8 (4KK7). 29. Address bits 0-3 clear bank busy (3SN7).	
T9	23. Destination tag 03 in delay rank 9 (4KK7). 24. Destination tag 03 in delay rank 9 sent to destination control-2 (6SV7).	30. Address bits 16, 17 in delay rank 9 (5XD7). 31. Address bits 0-3 in delay rank 9 (5XD7).	
T10	25. Destination tag 03 in delay rank 10, double error, parity or double error (6SV7).	32. Address bits 16, 17 in delay rank 10 (5XD7). 33. Address bits 0-3 in delay rank 10 (5XD7).	
T11	26. Double error-1 and SECDED error-1 signals sent to s/c register (6SV7).	34. Address information to s/c register (3SW7).	

# I/O READ SEQUENCE

Time	Control	Address	Data
	<ol style="list-style-type: none"> <li>1. Request received at ECS control and memory priority (6SP7).</li> <li>2. I/O tag to fanout (4LQ7).</li> <li>3. I/O request to memory priority translator (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>1. I/O tag and hold signal to I/O address holding register (6SP7).</li> <li>2. I/O address to holding register (3SO7).</li> </ol>	
T0	<ol style="list-style-type: none"> <li>4. I/O tag to SAS (3SL7, 3ST7).</li> <li>5. Go I/O (3SL7).</li> <li>6. Destination tag 05 and go address (3SL7).</li> <li>7. Destination tag 05 to SAS (3SL7, 3ST7).</li> <li>8. Accept sent to requesting port (6SU7).</li> </ol>	<ol style="list-style-type: none"> <li>3. I/O address to SAS (3ST7).</li> <li>4. Address flag from SAS tags and control (3SL7).</li> <li>5. Address bits 4-17 in memory reconfiguration and parity generator (4SX7).</li> <li>6. Address bits 0-3 in bank select (3SM7).</li> <li>7. Address bits 0-3 in refresh control (5SR7).†</li> <li>8. Transmit address bits 4-17 (4SX7).</li> <li>9. Go CSU (3SM7).</li> <li>10. Generate set bank busy (3SM7).</li> </ol>	
T1	<ol style="list-style-type: none"> <li>9. Destination tag 05 in delay rank 1 (4SJ7).</li> <li>10. I/O tag in delay rank 1 (4KK7).</li> </ol>	<ol style="list-style-type: none"> <li>11. Set bank busy (3SN7).</li> <li>12. Transmit go bank (3SN7) and address parity bit (3SW7).</li> <li>13. Address bits 16, 17 in delay rank 1 (5XD7).</li> <li>14. Address bits 0-3 in delay rank 1 (4KK7).</li> </ol>	
T2	<ol style="list-style-type: none"> <li>11. Destination tag 05 in delay rank 2 (5XD7).</li> <li>12. I/O tag in delay rank 2 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>15. Address bits 16, 17 in delay rank 2 (5XD7).</li> <li>16. Address bits 0-3 in delay rank 2 (5XD7).</li> </ol>	
T3	<ol style="list-style-type: none"> <li>13. Destination tag 05 in delay rank 3 (5XD7).</li> <li>14. I/O tag in delay rank 3 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>17. Address bits 16, 17 in delay rank 3 (5XD7).</li> <li>18. Address bits 0-3 in delay rank 3 (5XD7).</li> </ol>	
T4	<ol style="list-style-type: none"> <li>15. CSU address parity error in destination control-1 (delay rank 4) (4SJ7).</li> <li>16. CSU address parity error sent to s/c register.</li> <li>17. Destination tag 05 in delay rank 4 (5XD7).</li> <li>18. I/O tag in delay rank 4 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>19. Address bits 16, 17 in delay rank 4 (5XD7).</li> <li>20. Address bits 0-3 in delay rank 4 (5XD7).</li> </ol>	
T5	<ol style="list-style-type: none"> <li>19. Destination tag 05 in delay rank 5 (5XD7).</li> <li>20. I/O tag in delay rank 5 (5XD7).</li> <li>21. CSU address parity error in delay rank 5 (4KK7).</li> <li>22. CSU address parity error sent to destination control-1 (4SJ7).</li> <li>23. CMC input error sent to error sense circuit (4SY7).</li> </ol>	<ol style="list-style-type: none"> <li>21. Address bits 16, 17 in delay rank 5 (5XD7).</li> <li>22. Address bits 0-3 in delay rank 5 (5XD7).</li> </ol>	

† 5SR7 module is not present in AA120-C.

## I/O READ SEQUENCE (Cont'd)

Time	Control	Address	Data
T6	24. Destination tag 05 in delay rank 6 (4SJ7). 25. I/O tag in delay rank 6 (5XD7). 26. CSU address parity error in delay rank 6 (5XD7).	23. Address bits 16, 17 in delay rank 6 (5XD7). 24. Address bits 0-3 in delay rank 6 (5XD7).	
T7	27. Destination tag 05 in delay rank 7 (4SJ7). 28. Destination tag 05 in delay rank 7, full (enable to clear bank / busy) (4SJ7). 29. Full fanout (4LU7). 30. I/O tag in delay rank 7 (5XD7). 31. CSU address parity error in delay rank 7 (5XD7).	25. Address bits 16, 17 in delay rank 7 (5XD7). 26. Address bits 0-3 in delay rank 7 (5XD7). 27. Address bits 0-3 fanout (4LU7).	
T8	32. Destination tag 05 in delay rank 8 (4KK7). 33. I/O tag in delay rank 8 (5XD7). 34. CSU address parity error in delay rank 8 (5XD7).	28. Address bits 16, 17 in delay rank 8 (5XD7). 29. Address bits 0-3 in delay rank 8 (4KK7). 30. Address bits 0-3 clear bank busy (3SN7).	
T9	35. Destination tag 05 in delay rank 9 (4KK7). 36. Destination tag 05 in delay rank 9 sent to desination control-2 (6SV7). 37. I/O tag in delay rank 9 (5XD7). 38. CSU address parity error in delay rank 9 (5XD7).	31. Address bits 16, 17 in delay rank 9 (5XD7). 32. Address bits 0-3 in delay rank 9 (5XD7). 33. Address bit 3 in delay rank 9 (4KK7). 34. Address bit 3 (select CSU-0) fanout (3XE7, 4LU7).	
T10	39. Destination tag 05 in delay rank 10 (6SV7). 40. I/O tag in delay rank 10 (6SV7). 41. I/O tag and I/O read fanout (3XD7). 42. CSU address parity error to requesting port (6SV7). 43. Data on lines to requesting port (6SV7). 44. Double error, parity or double error (6SV7).	35. Address bits 16, 17 in delay rank 10 (5XD7). 36. Address bits 0-3 in delay rank 10 (5XD7).	1. CSU read data sent to holding register and syndrome generator (3XH7). 2. Select CSU (3XE7, 4LU7). 3. SECDED correction and data distribution (4X17). 4. Syndrome code bits 0-7 and SECDED error signal sent to desination control-2 (6SV7).
T11	45. Double error-1 and SECDED error-1 signals sent to s/c register (6SV7).	37. Address information to s/c register (3SW7).	5. Read data in holding register (3SS7). 6. I/O tag and I/O read signal gate read data (3SS7). 7. Transmit read data bits 0-59, transmission parity, and double-error to requesting port (3SS7).

# I/O WRITE SEQUENCE

Time	Control	Address	Data
	<ol style="list-style-type: none"> <li>1. Request received at ECS control and memory priority (6SP7).</li> <li>2. Write CM (3SO7).</li> <li>3. I/O tag to fanout (4LQ7).</li> <li>4. I/O request to memory parity transiator (4LM7).</li> <li>5. Shift SWA, SWB, SWC (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>1. I/O tag and hold signal to I/O address holding register (3SO7).</li> <li>2. I/O address to holding register (3SO7).</li> </ol>	<ol style="list-style-type: none"> <li>1. Write data received from requesting port (3SO7).</li> <li>2. I/O data parity check (6SQ7).</li> <li>3. CMC address/data parity error sent to s/e register (6SU7).</li> <li>4. CMC input error sent to requesting port (6SU7).</li> </ol>
T0	<ol style="list-style-type: none"> <li>6. I/O tag to SAS (3SL7, 3ST7).</li> <li>7. Go I/O (3SL7).</li> <li>8. Destination tag 07 and go address (3SL7).</li> <li>9. Destination tag 07 to SAS (3SL7, 3ST7).</li> <li>Aaccept sent to requesting port (6SU7).</li> </ol>	<ol style="list-style-type: none"> <li>3. I/O address to SAS (3ST7).</li> <li>4. Address flag from SAS tags and control (3SL7).</li> <li>5. Address bits 4-17 in memory reconfiguration and parity generator (4SX7).</li> <li>6. Address bits 0-3 in bank select (3SM7).</li> <li>7. Address bits 0-3 in refresh control (5SR7).†</li> <li>8. Transmit address bits 4-17 (4SX7).</li> <li>9. Go CSU (3SM7).</li> <li>10. Generate set bank busy (3SM7).</li> </ol>	<ol style="list-style-type: none"> <li>5. Write data bits 0-59 to I/O SWB (4HM7).</li> <li>6. SWB I/O write data bits 0-59 to SWC and SECDED code generator (3XH7).</li> </ol>
T1	<ol style="list-style-type: none"> <li>10. Destination tag 07 in delay rank 1 (4SJ7).</li> <li>11. Transmit write signal to CSU-0, -1 (4SJ7).</li> <li>12. I/O tag in delay rank 1 (4KK7).</li> </ol>	<ol style="list-style-type: none"> <li>11. Set bank busy (3SN7).</li> <li>12. Transmit go bank (3SN7) and address parity bit (3SW7).</li> <li>13. Address bits 16, 17 in delay rank 1 (5XD7).</li> <li>14. Address bits 0-3 in delay rank 1 (4KK7).</li> </ol>	<ol style="list-style-type: none"> <li>7. Write data bits 0-59 and SECDED code to SWD (3SW7).</li> <li>8. Transmit I/O write data and SECDED code (3SW7).</li> </ol>
T2	<ol style="list-style-type: none"> <li>13. Destination tag 07 in delay rank 2 (5XD7).</li> <li>14. I/O tag in delay rank 2 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>15. Address bits 16, 17 in delay rank 2 (5XD7).</li> <li>16. Address bits 0-3 in delay rank 2 (5XD7).</li> </ol>	
T3	<ol style="list-style-type: none"> <li>15. Destination tag 07 in delay rank 3 (5XD7).</li> <li>16. I/O tag in delay rank 3 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>17. Address bits 16, 17 in delay rank 3 (5XD7).</li> <li>18. Address bits 0-3 in delay rank 3 (5XD7).</li> </ol>	
T4	<ol style="list-style-type: none"> <li>17. CSU address parity error in destination control-1 (delay rank 4) (4SJ7).</li> <li>18. CSU address parity error sent to s/e register.</li> <li>19. Destination tag 07 in delay rank 4 (5XD7).</li> <li>20. I/O tag in delay rank 4 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>19. Address bits 16, 17 in delay rank (5XD7).</li> <li>20. Address bits 0-3 in delay rank 4 (5XD7).</li> </ol>	
T5	<ol style="list-style-type: none"> <li>21. Destination tag 07 in delay rank 5 (5XD7).</li> <li>22. I/O tag in delay rank 5 (5XD7).</li> <li>23. CSU address parity error in delay rank 5 (4KK7).</li> <li>24. CSU address parity error sent to destination control-1 (4SJ7).</li> <li>25. CMC input error sent to error sense circuit (4SY7).</li> </ol>	<ol style="list-style-type: none"> <li>21. Address bits 16, 17 in delay rank 5 (5XD7).</li> <li>22. Address bits 0-3 in delay rank 5 (5XD7).</li> </ol>	

† 5SR7 module is not present in AA120-C.

## I/O WRITE SEQUENCE (Cont'd)

Time	Control	Address	Data
T6	26. Destination tag 07 in delay rank 6 (4SJ7). 27. I/O tag in delay rank 6 (5XD7). 28. CSU address parity error in delay rank 6 (5XD7).	23. Address bits 16, 17 in delay rank 6 (5XD7). 24. Address bits 0-3 in delay rank 6 (5XD7).	
T7	29. Destination tag 07 in delay rank 7 (4SJ7). 30. Destination tag 07 in delay rank 7, full (enable to clear bank busy (4SJ7). 31. Full fanout (4LU7). 32. I/O tag in delay rank 7 (5XD7). 33. CSU address parity error in delay rank 7 (5XD7).	25. Address bits 16, 17 in delay rank 7 (5XD7). 26. Address bits 0-3 in delay rank 7 (5XD7). 27. Address bits 0-3 fanout (4LU7).	
T8	34. Destination tag 07 in delay rank 8 (4KK7). 35. I/O tag delay in rank 8 (5XD7). 36. CSU address parity error in delay rank 8 (5XD7).	28. Address bits 16, 17 in delay rank 8 (5XD7). 29. Address bits 0-3 in delay rank 8 (4KK7). 30. Address bits 0-3 clear bank busy (3SN7).	
T9	37. Destination tag 07 in delay rank 9 (4KK7). 38. Destination tag 07 in delay rank 9 sent to destination control-2 (6SV7). 39. I/O tag in delay rank 9 (5XD7). 40. CSU address parity error in delay rank 9 (5XD7).	31. Address bits 16, 17 in delay rank 9 (5XD7). 32. Address bits 0-3 in delay rank 9 (5XD7).	
T10	41. Destination tag 07 in delay rank 10 (6SV7). 42. CSU address parity error to requesting port (6SV7).	33. Address bits 16, 17 in delay rank 10 (5XD7). 34. Address bits 0-3 in delay rank 10 (5XD7).	
T11		35. Address information to s/c register (3SW7).	

# INCREMENT READ SEQUENCE

Time	Control	Address	Data
	<ol style="list-style-type: none"> <li>Set go increment to storage flag (4HG7).</li> <li>i delayed (4RO7).</li> <li>Go increment to storage to memory priority translator (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>Increment address formed in increment unit (3KJ7).</li> <li>Go increment to storage gates increment address (4HG7).</li> </ol>	
T0	<ol style="list-style-type: none"> <li>Destination tag 11-15 and go address (3SL7).</li> <li>Destination tag 11-15 to SAS (3SL7, 3ST7).</li> </ol>	<ol style="list-style-type: none"> <li>Increment address to SAS (3ST7).</li> <li>Address flag from SAS tags and control (5SL7).</li> <li>Address bits 4-17 in memory reconfiguration and parity generator (4SX7).</li> <li>Address bits 0-3 in bank select (3SM7).</li> <li>Address bits 0-3 in refresh control (5SR7).†</li> <li>Transmit address bits 4-17 (4SX7).</li> <li>Go CSU (3SM7).</li> <li>Generate set bank busy (3SM7).</li> </ol>	
T1	<ol style="list-style-type: none"> <li>Destination tag 11-15 in delay rank 1 (4SJ7).</li> <li>If no increment range error, increment read to X in delay rank 1 (4SJ7).</li> </ol>	<ol style="list-style-type: none"> <li>Set bank busy (3SN7).</li> <li>Transmit go bank (3SN7) and address parity bit (3SW7).</li> <li>Address bits 16, 17 in delay rank 1 (5XD7).</li> <li>Address bits 0-3 in delay rank 1 (4KK7).</li> </ol>	
T2	<ol style="list-style-type: none"> <li>Destination tag 11-15 in delay rank 2 (5XD7).</li> <li>Increment read to X in delay rank 2 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>Address bits 16, 17 in delay rank 2 (5XD7).</li> <li>Address bits 0-3 in delay rank 2 (5XD7).</li> </ol>	
T3	<ol style="list-style-type: none"> <li>Destination tag 11-15 in delay rank 3 (5XD7).</li> <li>Increment read to X in delay rank 3 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>Address bits 16, 17 in delay rank 3 (5XD7).</li> <li>Address bits 0-3 in delay rank 3 (5XD7).</li> </ol>	
T4	<ol style="list-style-type: none"> <li>CSU address parity error in destination control-1 (delay rank 4) (4SJ7).</li> <li>CSU address parity error sent to s/c register.</li> <li>Destination tag 11-15 in delay rank 4 (5XD7).</li> <li>Increment read to X in delay rank 4 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>Address bits 16, 17 in delay rank 4 (5XD7).</li> <li>Address bits 0-3 in delay rank 4 (5XD7).</li> </ol>	
T5	<ol style="list-style-type: none"> <li>Destination tag 11-15 in delay rank 5 (5XD7).</li> <li>Increment read to X in delay rank 5 (5XD7).</li> <li>CSU address parity error in delay rank 5 (4KK7).</li> <li>CSU address parity error sent to destination control-1 (4SJ7).</li> <li>CMC input error sent to error sense circuit (4SY7).</li> </ol>	<ol style="list-style-type: none"> <li>Address bits 16, 17 in delay rank 5 (5XD7).</li> <li>Address bits 0-3 in delay rank 5 (5XD7).</li> </ol>	

† 5SR7 module is not present in AA120-C.

INCREMENT READ SEQUENCE (Cont'd)

Time	Control	Address	Data
T6	21. Destination tag 11-15 in delay rank 6 (4SJ7). 22. Increment read to X in delay rank 6 (5XD7).	23. Address bits 16, 17 in delay rank 6 (5XD7). 24. Address bits 0-3 in delay rank 6 (5XD7).	
T7	23. Destination tag 11-15 in delay rank 7 (4SJ7). 24. CM to X tag (4RO7). 25. Read to X (4KK7). 26. Destination tag 11-15 in delay rank 7, full (enable to clear bank busy) (4SJ7). 27. Full fanout (4LU7). 28. Increment read to X in delay rank 7 (5XD7).	25. Address bits 16, 17 in delay rank 7 (5XD7). 26. Address bits 0-3 in delay rank 7 (5XD7). 27. Address bits 0-3 fanout (4LU7).	
T8	29. Destination tag 11-15 in delay rank 8 (4KK7). 30. Increment read to X in delay rank 8 (5XD7).	28. Address bits 16, 17 in delay rank 8 (5XD7). 29. Address bits 0-3 in delay rank 8 (4KK7). 30. Address bits 0-3 clear bank busy (3SN7).	
T9	31. Destination tag 11-15 in delay rank 9 (4KK7). 32. Set complement in rank 9 (6SV7). 33. Increment read to X in delay rank 9 (5XD7).	31. Address bits 16, 17 in delay rank 9 (5XD7). 32. Address bits 0-3 in delay rank 9 (5XD7). 33. Address bit 3 in delay rank 9 (4KK7). 34. Address bit 3 (select CSU-0) fanout (3XE7, 4LU7).	
T10	34. Destination tag 11-15 in delay rank 10 (6SV7). 35. Increment read to X in delay rank 10 (6SV7). 36. Read to X (6SV7). 37. Read to X fanout (3XE7). 38. Double error sent to holding register (3SS7). 39. Parity or double error sent to error sense circuit (4SY7).	35. Address bits 16, 17 in delay rank 10 (5XD7). 36. Address bits 0-3 in delay rank 10 (5XD7).	<ol style="list-style-type: none"> <li>1. CSU read data sent to holding register and syndrome generator (3XH7).</li> <li>2. Select CSU (3XE7, 4LU7).</li> <li>3. SECDED correction and data distribution (4XI7).</li> <li>4. Read data bits 0-59 sent to X register (4XI7).</li> <li>5. Syndrome code bits 0-7 and SECDED error signal sent to destination control-2 (6SV7).</li> </ol>
T11	40. Double error-1 and SECDED error-1 sent to s/c register (6SV7).	37. Address information sent to s/c register (3SW7).	

# INCREMENT WRITE SEQUENCE

Time	Control	Address	Data
	<ol style="list-style-type: none"> <li>1. Set go increment to storage flag (4HG7).</li> <li>2. 1 delayed (4RO7).</li> <li>3. Go increment to storage to memory priority translator (4LM7).</li> <li>4. Shift SWA, SWB, SWC (4LM7).</li> </ol>	<ol style="list-style-type: none"> <li>1. Increment address formed in increment unit (3KJ7).</li> <li>2. Go increment to storage gates increment address (4HG7).</li> </ol>	<ol style="list-style-type: none"> <li>1. Xi register data to SWA (4HM7).</li> </ol>
T0	<ol style="list-style-type: none"> <li>5. Destination tag 16 or 17 to SAS (3SL7, 3ST7).</li> <li>6. Destination tag 16 or 17 and go address (3SL7).</li> </ol>	<ol style="list-style-type: none"> <li>3. Increment address to SAS (3ST7).</li> <li>4. Address flag from SAS tags and control (3SL7).</li> <li>5. Address bits 4-17 in memory reconfiguration and parity generator (4SX7).</li> <li>6. Address bits 0-3 in bank select (3SM7).</li> <li>7. Address bits 0-3 in refresh control (5SR7).†</li> <li>8. Transmit address bits 4-17 (4SX7).</li> <li>9. Go CSU (3SM7).</li> <li>10. Generate set bank busy (3SM7).</li> </ol>	<ol style="list-style-type: none"> <li>2. Xi register data in SWB (4HM7).</li> <li>3. Xi register data in SWC and SECEDED code generator (3XH7).</li> </ol>
T1	<ol style="list-style-type: none"> <li>7. Destination tag 16 or 17 delay rank 1 (4SJ7).</li> <li>8. Transmit write signal to CSU-0, -1 (4SJ7).</li> </ol>	<ol style="list-style-type: none"> <li>11. Set bank busy (3SN7).</li> <li>12. Transmit go bank (3SN7) and address parity bit (3SW7).</li> <li>13. Address bits 16, 17 in delay rank 1 (5XD7).</li> <li>14. Address bits 0-3 in delay rank 1 (4KK7).</li> </ol>	<ol style="list-style-type: none"> <li>4. Xi register data in SWD (3SW7).</li> <li>5. Transmit write data to CSU-0.</li> </ol>
T2	<ol style="list-style-type: none"> <li>9. Destination tag 16 or 17 in delay rank 2 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>15. Address bits 16, 17 in delay rank 2 (5XD7).</li> <li>16. Address bits 0-3 in delay rank 2 (5XD7).</li> </ol>	
T3	<ol style="list-style-type: none"> <li>10. Destination 16 or 17 in delay rank 3 (5XD7).</li> <li>11. CSU address parity error in destination control-1 (delay rank 4)</li> </ol>	<ol style="list-style-type: none"> <li>17. Address bits 16, 17 in delay rank 3 (5XD7).</li> <li>18. Address bits 0-3 in delay rank 3 (5XD7).</li> </ol>	
T4	<ol style="list-style-type: none"> <li>12. CSU address parity error sent to s/c register.</li> <li>13. Destination tag 16 or 17 in delay rank 4 (5XD7).</li> </ol>	<ol style="list-style-type: none"> <li>19. Address bits 16, 17 in delay rank 4 (5XD7).</li> <li>20. Address bits 0-3 in delay rank 4 (5XD7).</li> </ol>	
T5	<ol style="list-style-type: none"> <li>14. Destination tag 16 or 17 in delay rank 5 (5XD7).</li> <li>15. CSU address parity error in delay rank 5 (4KK7).</li> <li>16. CSU address parity error sent to destination control-1 (4SJ7).</li> <li>17. CMC input error sent to error sense circuit (4SY7).</li> </ol>	<ol style="list-style-type: none"> <li>21. Address bits 16, 17 in delay rank 5 (5XD7).</li> <li>22. Address bits 0-3 in delay rank 5 (5XD7).</li> </ol>	

† 5SR7 module is not present in AA120-C.

INCREMENT WRITE SEQUENCE (Cont'd)

Time	Control	Address	Data
T6	18. Destination tag 16 or 17 in delay rank 6 (4SJ7).	23. Address bits 16, 17 in delay rank 6 (5XD7). 24. Address bits 0-3 in delay rank 6 (5XD7).	
T7	19. Destination tag 16 or 17 in delay rank 7 (4SJ7). 20. Destination tag 16 or 17 in delay rank 7, full (enable to clear bank busy) (4SJ7). 21. Full fanout (4LU7).	25. Address bits 16, 17 in delay rank 7 (5XD7). 26. Address bits 0-3 in delay rank 7 (5XD7). 27. Address bits 0-3 fanout (4LU7).	
T8	22. Destination tag 16 or 17 in delay rank 8 (4KK7).	28. Address bits 16, 17 in delay rank 8 (5XD7). 29. Address bits 0-3 in delay rank 8 (4KK7). 30. Address bits 0-3 clear bank busy (3SN7).	
T9	23. Destination tag 16 or 17 in delay rank 9 (4KK7). 24. Destination tag 16 or 17 to destination control-2 (6SV7).	31. Address bits 16, 17 in delay rank 9 (5XD7). 32. Address bits 0-3 in delay rank 9 (5XD7).	
T10	25. Destination tag 16 or 17 in delay rank 10 (6SV7).	33. Address bits 16, 17 in delay rank 10 (5XD7). 34. Address bits 0-3 in delay rank 10 (5XD7).	
T11		35. Address information sent to s/c register (3SW7).	

## DETAILED PAK DIAGRAM (PPS 3.14)

### WRITE PYRAMID AND CM ADDRESS

#### WRITE PYRAMID

Each stage of the write pyramid (PPS 3.14) loads data from the previous stages as well as one 12-bit PP word from the FD register upon enabling by control signals from the BH module. The act of setting one of the D1-D4 full flip-flops, as accomplished on CM write instructions (62 and 63), disables data transfers to the stage so designated as being full. Once bits 00-59 are assembled in D4 and D5, the write pyramid output register transmits to CMC. D4 is held until accepted by the write pyramid output register and transmitter because D4FULL resets only on BGWOUT.

The BG control translator and related circuitry within the BG module produce control signals for PPS-CMC intercommunication.

Signals BQEXCH, WROUT and BURREQ are developed in the K translators to indicate an exchange jump (26), write to CM (60, 61) or read from CM (62, 63). The REQ PRI1 signal is a necessary condition for acceptance of any of these requests for communication with CMC. REQ PRI1 exists on conditions as shown in the BG control translator. The combination of REQ PRI1 and BQEXCH or WROUT produces BGAO which, in turn, sets the BUSY1 flip-flop.

#### PPS-CMC CONTROL SIGNALS

The BG output register and transmitter transmits the BGCMRQ signal in the form of PPORQ to CMC. This signal arises whenever the BG pak accepts a CM read or write request, and commands the CMC to interpret the bits transmitted from the write address output register as the central memory address bits for a data transfer. These address bits (PPOA00-17) are ignored by CMC unless accompanied by PPORQ.

The BG output transmitter also sends a PPOWR signal to CMC indicating a write instruction. If this is not sent with PPORQ, a read CM request is assumed by CMC.

On an exchange jump instruction, the BG output transmitter sends control bits indicating which central processor will be interrupted (not sent in a single CPU system) and defining which type of exchange jump is required. PPOXNO means CPU-1 is involved. PPOMXN means a monitor exchange jump to A (261 instruction). PPOMAN means a monitor exchange jump to MA (262 instruction). PPOIXNQ, PPOMAN means an exchange jump (260 instruction).

When CMC is ready for another request, it sends the accept signal which clears the BUSY1 flip-flop, allowing another request to generate the BGAO signal.

In the interval between the setting of the BUSY1 flip-flop and the reception of the accept signal, REQ PRI2 replaces REQ PRI1 (see translation in BG control translator). When another request occurs under these circumstances the BUSY1 flip-flop sets, inhibiting the BGLPRI (latch priority) signal.

This situation is depicted in figure 4-2-7 for a series of CMC write requests (a write request from the K9 decoder = WROUT). With the BUSY1 flip-flop initially set by the first write request, the BUSY1 flip-flop sets on a second request. The BUSY1 flip-flop inhibits further generation of the BGLPRI signal. BGLPRI does two things in the CV module. First, it sets the BGRQPR signal and, second, it latches the PP count into the priority PP holding flip-flop. With BGLPRI inhibited, the number of the PP to make the second request is locked into the priority PP holding flip-flop, making it the priority PP (this is PP1 in figure 4-2-7). When the first PP's request is accepted (ACCEPT) as shown in figure 4-2-8, BUSY1 clears and enables the priority PP to request central memory access. When the priority PP is in the pre-slot, the comparison of the priority PP holding flip-flop and the CV incrementer yields equality. The equality signal produces BGRQPR in PP1 slot time to enable PP1's second attempt at a write request.

#### READ REQUESTS

When a read request (BURREQ) is received and accepted by the BG control translator, the REQ CONTROL signal exists. REQ CONTROL then successively sets the second outstanding request flip-flop (2nd OSRFF) and first outstanding request flip-flop (1st OSRFF). If the first outstanding request flip-flop was not set initially, the 2nd OSRFF resets at the end of this sequence. The 1st OSRFF may be reset only by C5C4.

If another read request arises before the 1st OSRFF resets, the latch PP request flip-flop inhibits the BGLRQ2 signal. In a manner similar to BGLPRI, BGLRQ2 latches the second PP (making a read) request. It does this by holding the RQ2 PP holding FF (CV module - PPS 3.14) at the count that was in the CV counter when BGLRQ2 occurred. When the CV counter equals the value in the RQ2 PP holding FF (every 500 ns when the same PP is in the slot) CVC5EN will be generated from the comparator, inhibiting the locked out PP from re-

ceiving data from the read pyramid (PPS 3.13). The purpose of this arrangement is to ensure that the second requesting PP does not read data intended for the first requesting PP.

On a read request, BGAO is not generated if the 2nd OSRFF is set or if 1st OSRFF . BKC5FL. This prevents CMC from returning data to the PPS before the read pyramid is ready for it.

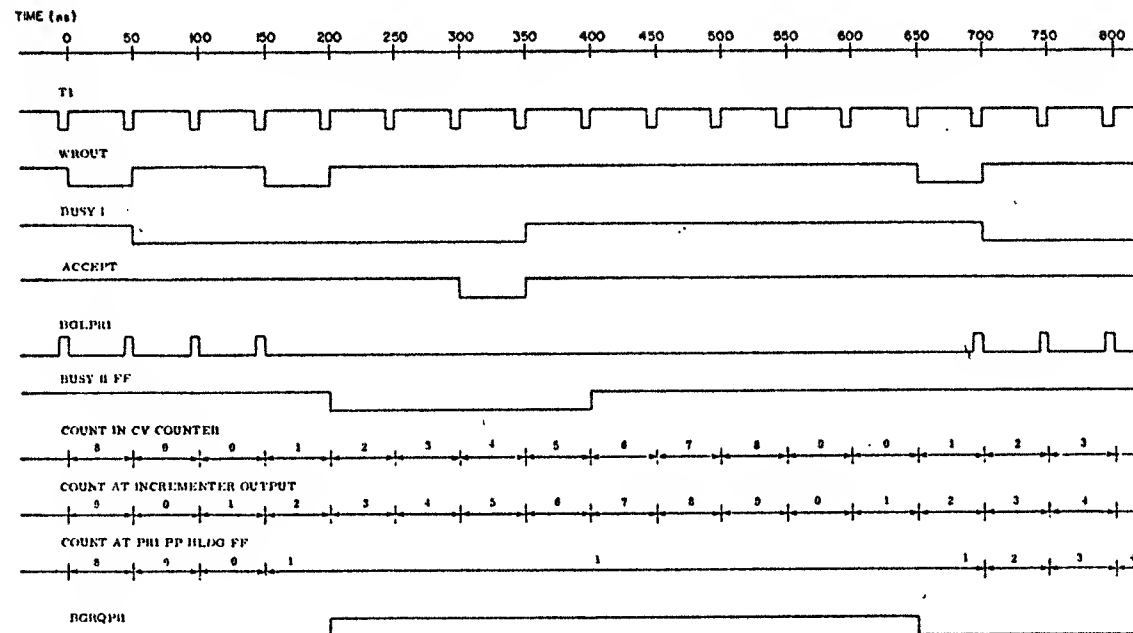
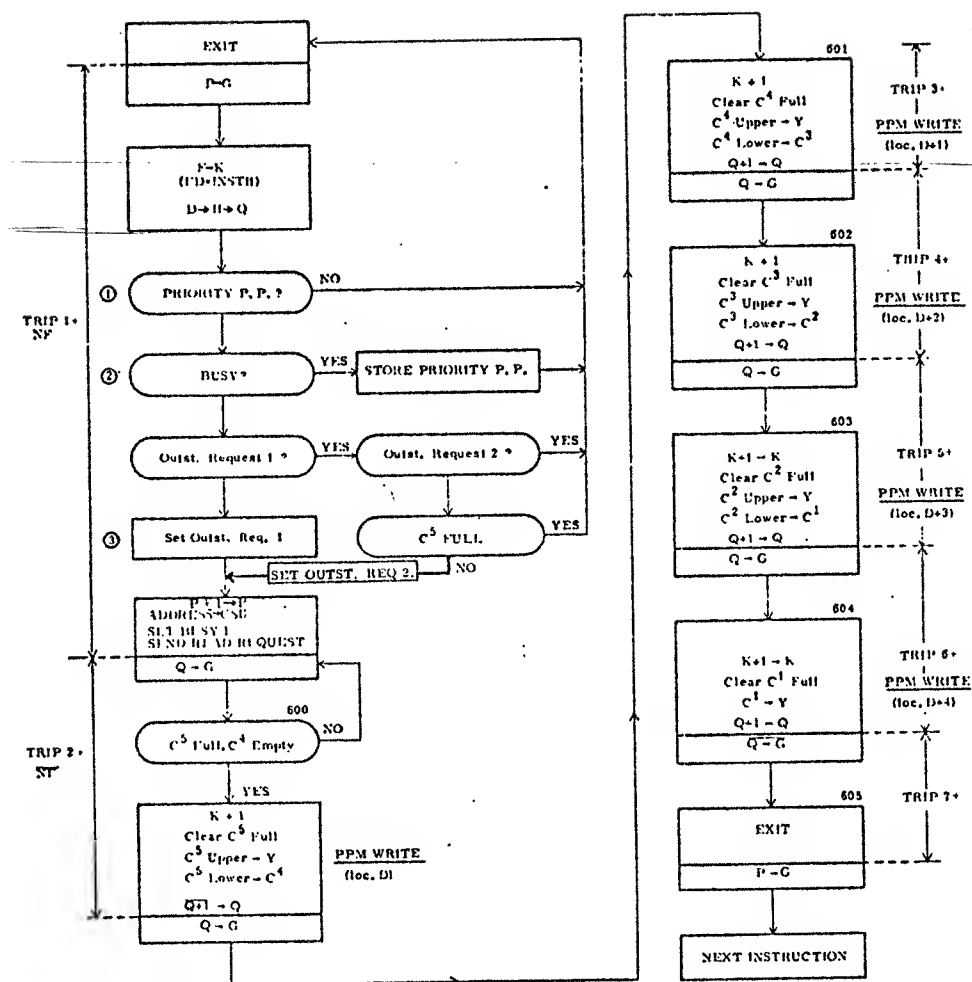


Figure 4-2-7. PPS - CMC Priority



Code	Name
CRD 60	Read CM, loc. A to D, D+1, D+2, D+3, D+4.

Note: ① Priority P, P, defined as a P, P, whose number corresponds to the Priority Latch.  
 ② 'Busy' is cleared by 'Accept'.  
 ③ 'Outstanding Request' closed by 'Data Ready'.

Figure 5-A-11. PPS Instruction 60

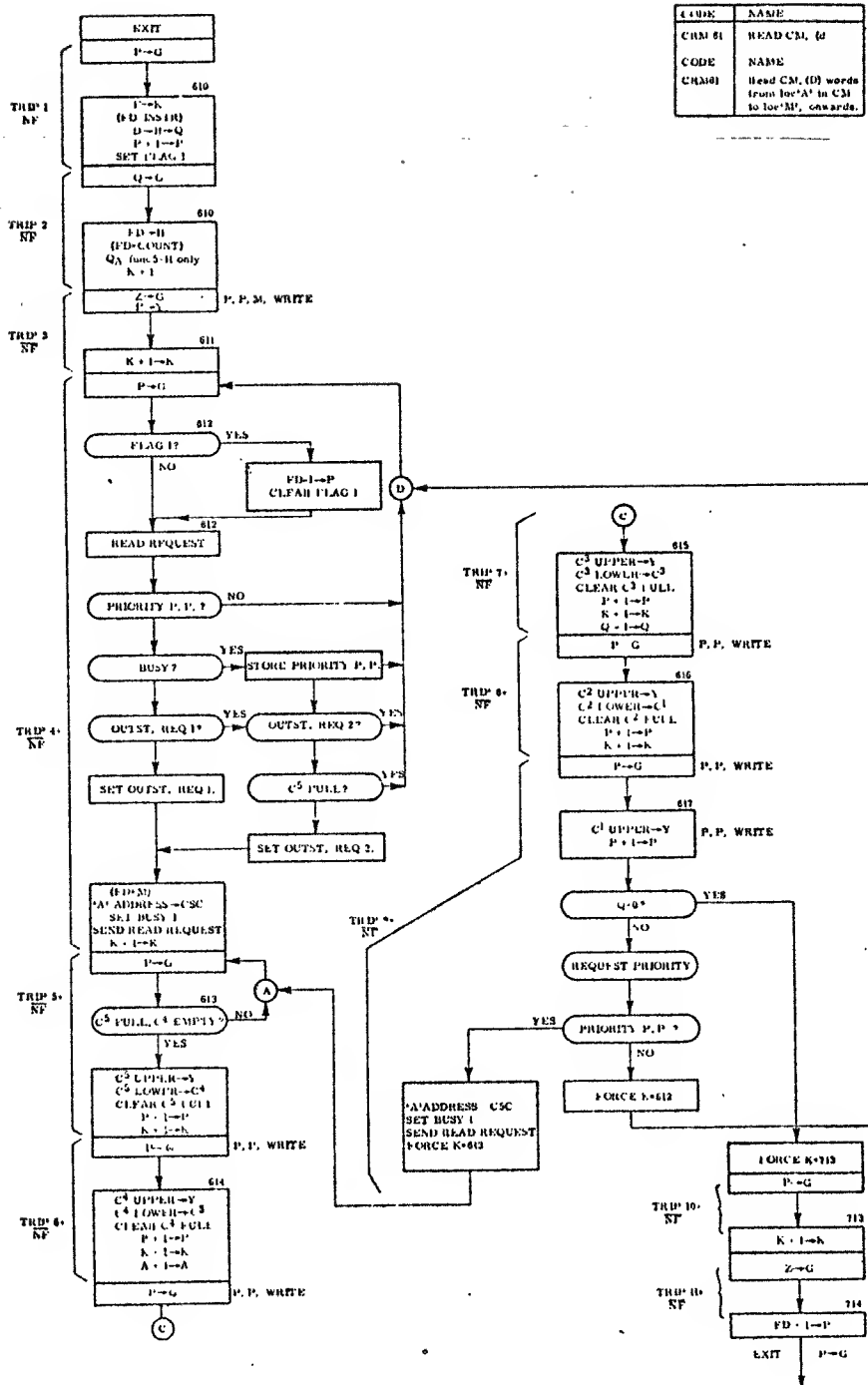


Figure 5-A-12. PPS Instruction 61

Label	Meaning
WRITE	Write a 16 word from D <sub>0</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> to a 160 'M'.

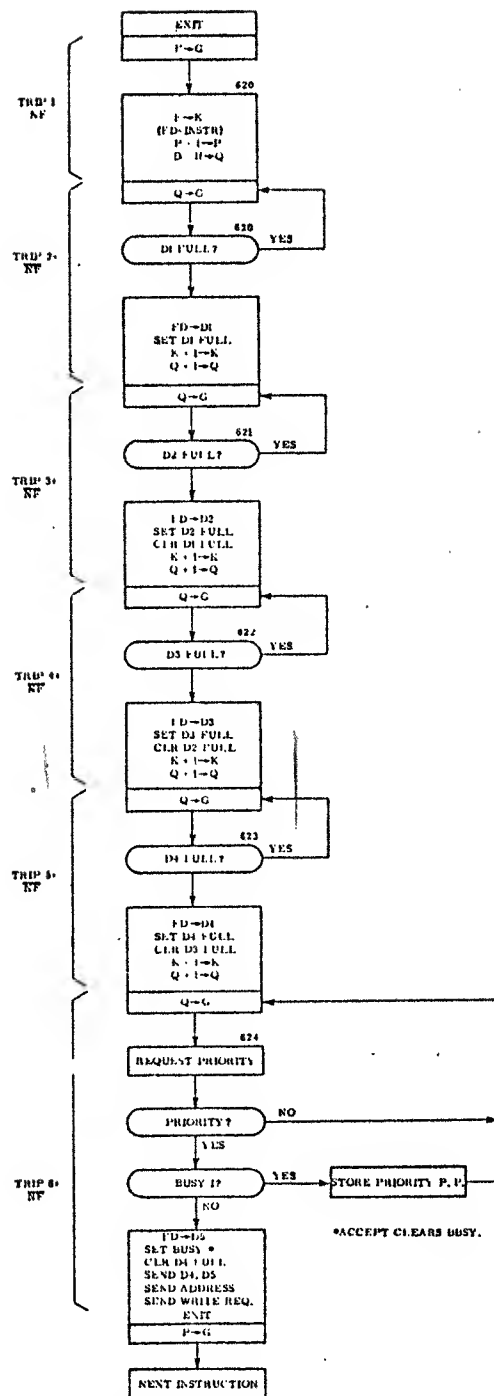


Figure 5-A-13. PPS Instruction 62

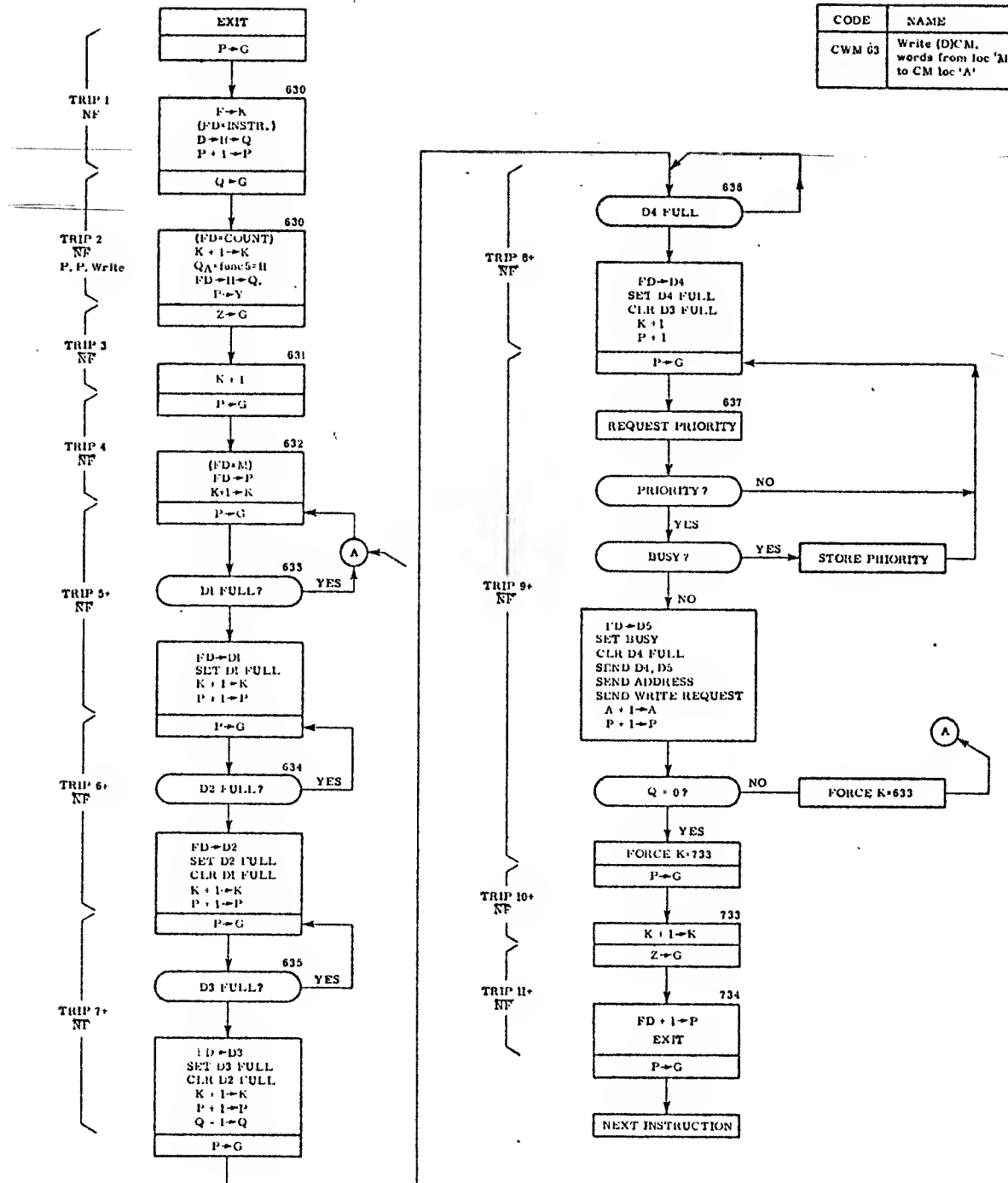


Figure 5-A-14. PPS Instruction 63



TABLE 5-2-15. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF CLEAR

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Illegal instruction	<ol style="list-style-type: none"> <li>1. Execute the illegal instruction as if it were a pass.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Execute the illegal instruction as if it were a pass.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>
Exit condition bit 48 set by an increment read of an address out of range	<ol style="list-style-type: none"> <li>1. Read all zeros to the selected X register.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Read all zeros to the selected X register.</li> <li>2. Continue execution.</li> </ol>
Exit condition bit 48 set due to an increment write of an address out of range	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Continue execution.</li> </ol>
Exit condition bit 48 set due to an RNI or branch address out of range	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at RAC.</li> <li>3. Clear P.</li> <li>4. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at RAC.</li> <li>3. Clear P.</li> <li>4. Exchange jump to MA and set MF.</li> </ol>
Exit condition bit 48 set on CMC instruction a. C1 or C2 > 9 b. K1 or K2 address out of range	<ol style="list-style-type: none"> <li>1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range.</li> <li>2. Continue with next 60-bit instruction.</li> </ol>

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Exit condition bit 48 set by an ECS address range check	<ol style="list-style-type: none"> <li>1. Force ECS instruction to execute as a pass instruction.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Force ECS instruction to execute as a pass instruction.</li> <li>2. Continue execution with next 60-bit word.</li> </ol>
Infinite condition (bit 49) Indefinite condition (bit 50) ECS flag register parity (bit 51) CMC input error condition (bit 52) CM data error condition (bit 53)	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at RAC.</li> <li>3. Clear P.</li> <li>4. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Continue execution.</li> </ol>
CMC input error condition (bit 52)	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Continue execution.</li> </ol>
00 instruction	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at RAC.</li> <li>3. Clear P.</li> <li>4. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at RAC.</li> <li>3. Clear P.</li> <li>4. Exchange jump to MA and set MF.</li> </ol>
Breakpoint signal from CMC (refer to breakpoint notes)	<ol style="list-style-type: none"> <li>1. Execute remaining parcels of 60-bit word currently executing.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>	<ol style="list-style-type: none"> <li>1. Execute remaining parcels of 60-bit word currently executing.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at RAC.</li> <li>4. Clear P.</li> <li>5. Exchange jump to MA and set MF.</li> </ol>

TABLE 5-2-14. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF SET

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Illegal Instruction	<ol style="list-style-type: none"> <li>1. Execute the illegal instruction as if it were a pass.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Execute the illegal instruction as if it were a pass.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>
Exit condition bit 48 set by an increment read of an address out of range	<ol style="list-style-type: none"> <li>1. Read all zeros to the selected X register.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Read all zeros to the selected X register.</li> <li>2. Continue execution.</li> </ol>
Exit condition bit 48 set by an increment write of an address out of range	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Continue execution.</li> </ol>
Exit condition bit 48 set on RNI or branch out of range	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at IAC.</li> <li>3. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at IAC.</li> <li>3. Clear P.</li> </ol>
Exit condition bit 48 set on CMU instruction a. C1 or C2 > 9 b. K1 or K2 address out of range	<ol style="list-style-type: none"> <li>1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range.</li> <li>2. Continue with next 60-bit instruction.</li> </ol>

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Exit condition bit 48 set by an ECS address range check	<ol style="list-style-type: none"> <li>1. Force ECS instruction to execute as a pass instruction.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Force ECS instruction to execute as a pass instruction.</li> <li>2. Exit to next 60-bit word.</li> <li>3. Continue execution with next 60-bit word.</li> </ol>
Infinite condition (bit 49) Indefinite condition (bit 50) ECS flag register parity (bit 51) CMC input error condition (bit 52) CM data error condition (bit 53)	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at IAC.</li> <li>3. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Continue execution.</li> </ol>
CMC input error condition (bit 52)	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Block write operation, contents of CM is unchanged.</li> <li>2. Continue execution.</li> </ol>
60 instruction	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at IAC.</li> <li>3. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Stop CPU.</li> <li>2. Store P and exit condition bits at IAC.</li> <li>3. Clear P.</li> </ol>
Breakpoint signal from CMC (refer to breakpoint notes)	<ol style="list-style-type: none"> <li>1. Execute remaining parcels of 60-bit word currently executing.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>	<ol style="list-style-type: none"> <li>1. Execute remaining parcels of 60-bit word currently executing.</li> <li>2. Stop CPU.</li> <li>3. Store P and exit condition bits at IAC.</li> <li>4. Clear P.</li> </ol>

## DETAILED PAK DIAGRAM (CPU 3.19)

### RETURN JUMP SEQUENCE

The return jump sequence controls operations necessary to perform the following instructions:

00xxx	Monitor Stop (Error Exit to MA)
010xK	Return Jump to K
013jK	Central Exchange Jump

#### RETURN JUMP 010

The 010 instruction stores an unconditional jump instruction (0400) to the current program address plus one ( $P + 1$ ) in the upper half of memory location  $K + RA$ , then branches to  $K + 1 + RA$  for the next instruction.

A jump to an address out of range enables the AOR sequence (CPU 3.17). The CPU response is dependent on whether the appropriate exit mode selection was made and the monitor flag /MEJ/CEJ condition.

#### CENTRAL EXCHANGE JUMP 013

The 013 instruction is enabled or disabled by the MEJ/CEJ switch on the dead start panel. If the switch is enabled, the return jump sequence allows the processor to send an exchange request (EXJREQ) to CMC. CMC then responds with request exchange (REQEXJ), which enables the exchange jump sequence (CPU 3.20) and unconditionally exchange jumps the CPU, regardless of the state of the monitor flag bit. However, instruction action differs depending on whether the monitor flag bit is set or clear:

##### Monitor Flag clear

The exchange sequence (CPU 3.20) makes the starting address for the exchange from the 18-bit monitor address register (MA). During the exchange, the monitor flag bit is set.

##### Monitor Flag set

The exchange sequence (CPU 3.20) takes the 18-bit starting address formed by adding K to the contents of Bj during the return jump sequence. During the exchange, the monitor flag bit is cleared.

If the MEJ/CEJ switch is in the disable position, the 013 instruction is illegal. The return jump sequence detects the illegal condition. NSETIL sets the illegal FF (CPU 3.27). The illegal FF, in turn, sets the error exit FF (CPU 3.17). Error exit clears the U3 instruction register and forces a return jump error exit sequence.

#### MONITOR STOP (Error Exit to MA) 00

The 00 instruction is enabled or disabled by the MEJ/CEJ switch on the dead start panel. The return jump sequence is enabled by an instruction decode of 00, or by an error exit that caused the U3 register to be cleared and thus forced an instruction decode of 00. With the MEJ/CEJ switch in the disable position, the processor has no central exchange or monitor exchange jump capability, so the return jump sequence clears the run FF and stops the processor.

In the enable position, the processor has the exchange jump capability, so the 00 instruction is executed in two passes through the return jump sequence.

The first pass records at RA a monitor stop instruction (00), the exit condition bits (EE), and the program address at exit time in the following format:

##### Store at RA

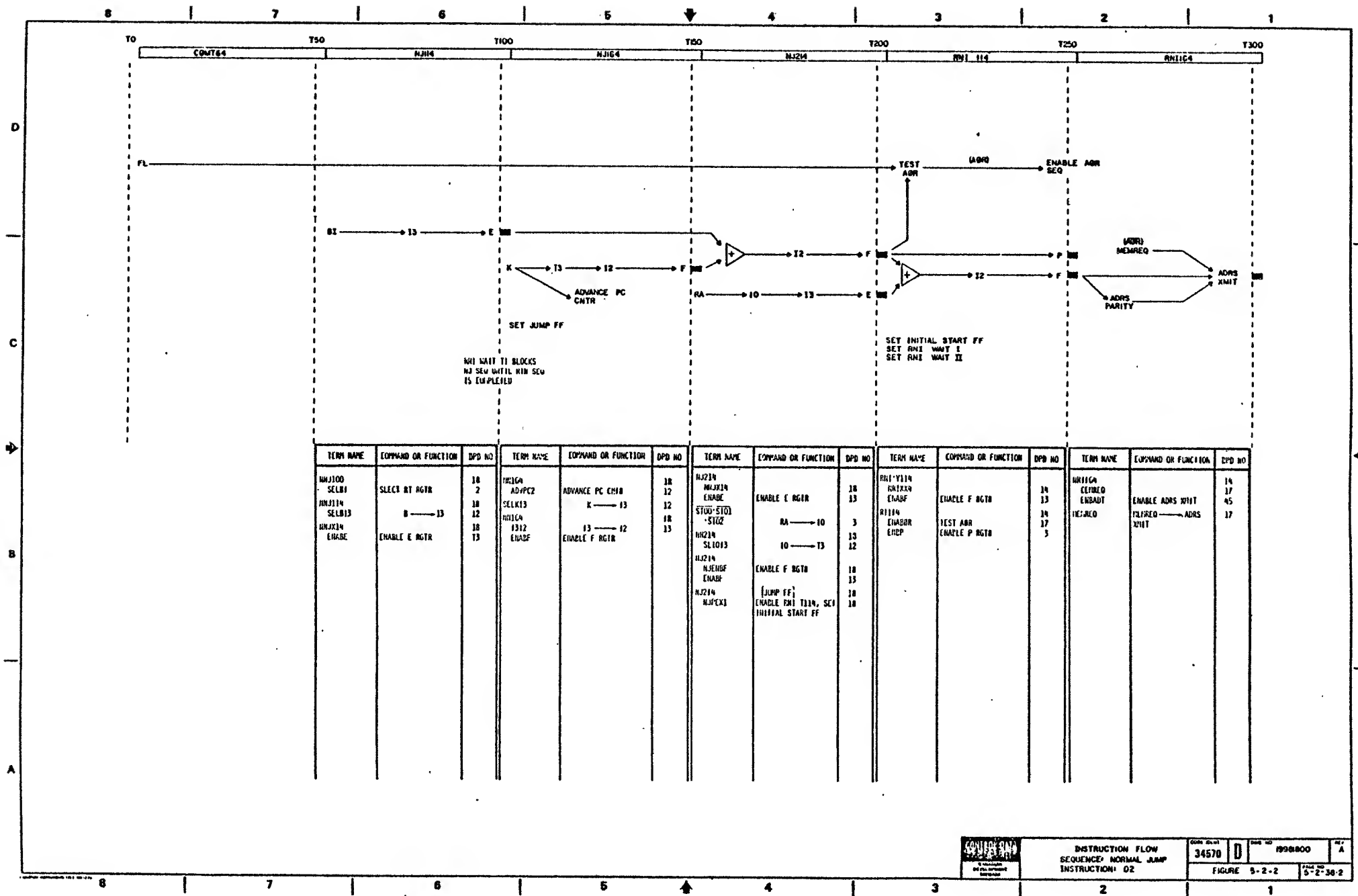
00	00	xxxxxx	0000000000
	Error P Register		
	bits	(Program Address)	

The return jump sequence then waits until CMC responds with an accept. Since the return jump wait FF is set, accept allows setting the RNI initial start FF by generating NRTJEX (CPU 3.16). The RNI sequence, using the P register cleared to zeros during the first return jump pass, reads the 00 instruction at RA. The 00 instruction enables the return jump sequence for the second pass. The second pass generates exchange request (EXJREQ) if the monitor flag is clear. If the monitor flag is set, the run FF is cleared and the CPU stops.

The error response conditions with MEJ/CEJ enabled and monitor flag set or clear, or MEJ/CEJ disabled, are detailed tables 5-2-14 and 5-2-15.







# DETAILED PAK DIAGRAM (CPU 3.18)

## NORMAL JUMP SEQUENCE

The normal jump sequence controls the operations necessary to perform the following instructions:

02ixk	Jump to (Bi) + K
030jk	Branch to K if (Xj) = 0
031jk	Branch to K if (Xj) ≠ 0
032jk	Branch to K if (Xj) Positive
033jk	Branch to K if (Xj) Negative
034jk	Branch to K if (Xj) in Range
035jk	Branch to K if (Xj) Out of Range
036jk	Branch to K if (Xj) Definite
037jk	Branch to K if (Xj) Indefinite
04ijk	Branch to K if (Bi) = (Bj)
05ijk	Branch to K if (Bi) ≠ (Bj)
06ijk	Branch to K if (Bi) ≥ (Bj)
07ijk	Branch to K if (Bi) < (Bj)

The 02 instruction performs an unconditional jump to the address specified by the contents of index register Bi, plus the K portion of the instruction. The branch address is K when i = 0.

The conditional jump instructions 030-037, 04-07 branch to address K if the jump condition specified by the instruction is met. Jump conditions are defined as follows:

02	Unconditional
030	Xj = All "1's" or "0's" (60 bits)
031	Xj ≠ All "1's" or "0's" (60 bits)
032	XSR1 (Xj Positive)
033	XSR1 (Xj Negative)
034	Xj exp ≠ 3777 + 4000 (Xj in Range)
035	Xj exp = 3777 + 4000 (Xj Out of Range)
036	Xj exp ≠ 1777 + 6000 (Xj Definite)
037	Xj exp = 1777 + 6000 (Xj Indefinite)
04	Bi = Bj All Pass + EAC
05	Bi ≠ Bj All Pass + EAC
06	Bi ≥ Bj BSR1 . BSR2 + BSR1 = BSR2 . F NEGATIVE
07	Bi < Bj BSR1 . BSR2 + BSR1 = BSR2 . F POSITIVE

A successful test of a specified jump condition effects an RNI initial start operation. The next instruction is executed at address K after RA has been added. An unsuccessful test of the jump condition causes a sequence exit which effects an RNI for the next instruction in the current program.

A jump to an address out of range enables the AOR sequence (CPU 3.17). The CPU response is dependent on whether the appropriate exit mode selection was made and the monitor flag /MEJ/CEJ condition.

## DETAILED PAK DIAGRAM (CPU 3.17)

### AOR SEQUENCE

The AOR sequence timing located on the GL module is enabled when an address out of range condition is detected. Address out of range may be detected as a result of one of the following conditions:

1. Increment read address out of range
2. Increment write address out of range
3. RNI or branch out of range
4. Compare/move instruction with:
  - (a) C1 or C2 > 9, or
  - (b) K1 or K2 address out of range
5. ECS address out of range check.

The AOR sequence aborts the out of range memory request, the compare/move, or ECS instruction and sets the clear CR9 FF. This allows for the transfer of zeros from CR9 to the selected X register when an address out of range occurs on an increment read instruction. Blocking the memory request also prevents a write operation on an increment write instruction. Address out of range sets bit 48 of the error exit register (CPU 3.4). If the corresponding exit mode register selection was also made, exit condition sensed (ECONDS) enables setting the error exit FF at sequence exit time.

### ERROR EXIT RESPONSES

The error exit FF, also located on the GL module, is set at sequence exit time when one of the following error conditions is detected:

1. Exit condition sensed (ECONDS)
  - (a) Bit 48 Address out of range
  - (b) Bit 49 Infinite condition
  - (c) Bit 50 Indefinite condition
  - (d) Bit 51 Flag operation parity error condition
  - (e) Bit 52 CMC input parity error condition
  - (f) Bit 53 CM data parity error condition
2. Illegal instruction
3. Breakpoint sensed
4. 30-bit Illegal FF.

Setting of the error exit FF clears the U3 instruction register and enables a return jump error exit sequence (CPU 3.19).

## DETAILED PAK DIAGRAM (CPU 3.16)

### ACCEPT SEQUENCE

The accept sequence consists of a timing chain and control that are enabled by CMC data ready (DARDY). Data ready is sent 50 ns ahead of output data from CMC. The accept sequence generates control signals that enable the acceptance of data.

NDR50 selects operand register XI for receipt of a data operand as the result of an increment read memory reference.

DR64 enables an RNI initial start sequence exit (FORCEX) after receipt of a new 60-bit instruction word.

CMDRM is used by the compare/move control. It indicates receipt of a data word to allow start of the data sequence.

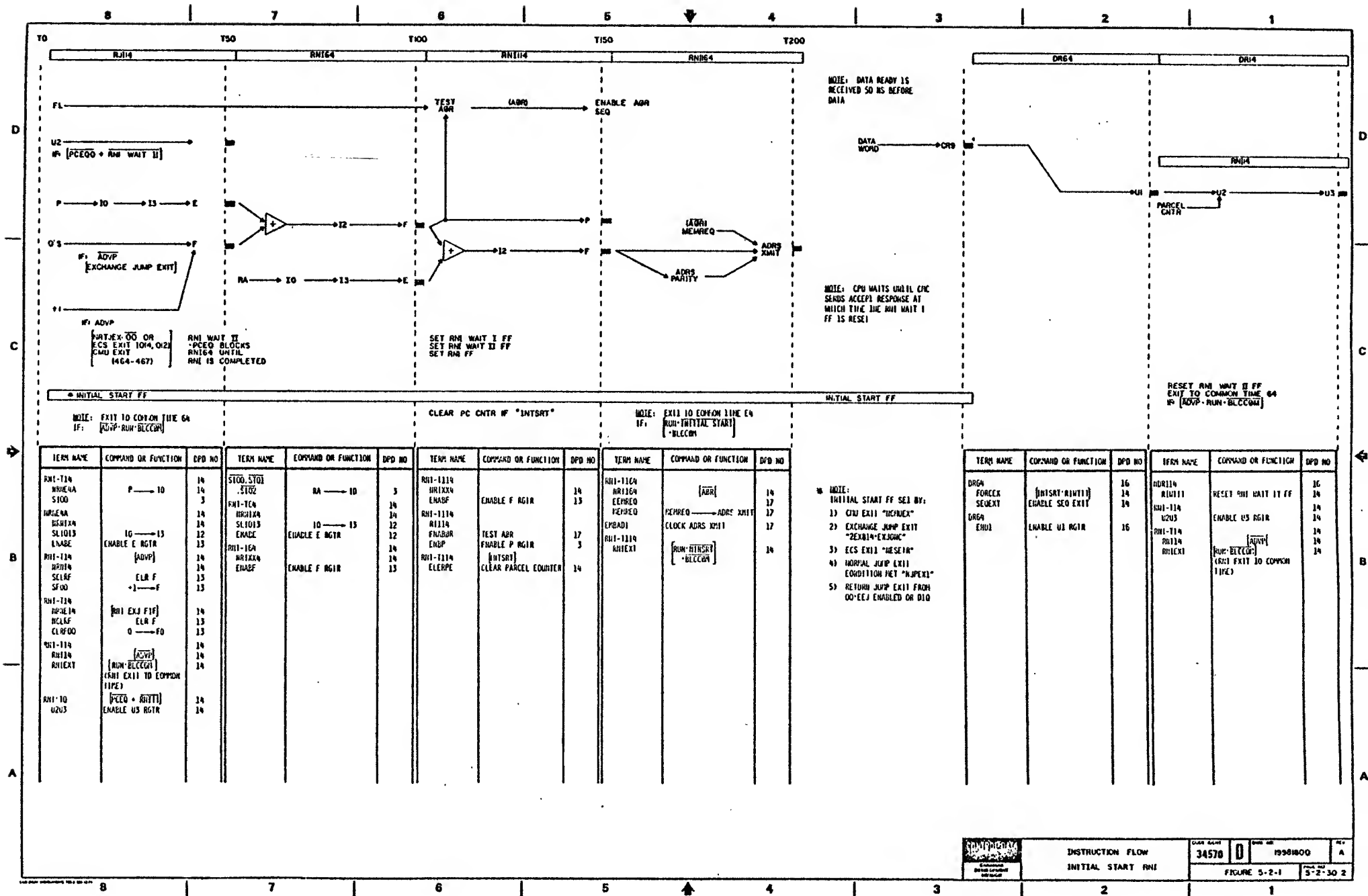
When an address out of range fault occurs, NARIH from the AOR sequence forces the start of a false accept sequence. Setting of the CLR CR9 FF (CPU 3.17) during AOR ensure that zeros are gated to the CR9 register.

## DETAILED PAK DIAGRAM (CPU 3.15)

### COMMON TIME SEQUENCE

The common time sequence controls the initial operating conditions of each of the instruction sequences. Common time 50/64 is initiated by RNIEXT, and provides instruction translation and decoding operations to initiate the appropriate control sequence for execution. Common time 0/14 is initiated by COMEXT, and provides controls to return the results of an instruction to a selected B or X register.

The common time sequence timing chain is contained on the GT module. At common time 50, ADVPC1 is generated to update the parcel counter pointing to the next 15-bit instruction. Common time 64 (COMT64) enables the function decode translator circuits located on the GU module. The GU module provides a decode of the instruction in U3, and generates the appropriate GO signal to the control sequence for the instruction type decoded.



## DETAILED PAK DIAGRAM (CPU 3.14)

### RNI SEQUENCE

The RNI sequence controls the operations necessary to initially start the CPU following an exchange jump, and performs RNIs for subsequent instructions to be executed.

RNIs are referred to as an initial start RNI, a full RNI, or a parcel RNI. An initial start RNI is enabled by the following conditions:

- |                    |                           |
|--------------------|---------------------------|
| 1. EXJONC . 2EX814 | Exchange Jump Exit        |
| 2. NESETR          | Normal ECS Exit           |
| 3. NJPEX1          | Normal Jump Condition Met |
| 4. NRTJEX          | Return Jump Exit          |
| 5. NCMUEX          | CMU Exit                  |

An initial start RNI obtains a new 60-bit instruction word from central memory. A full RNI is similar in operation to an initial start RNI in that a new 60-bit instruction word is obtained from central memory; however, a full RNI is initiated between execution of the first and

second instructions of the word being processed. A parcel RNI obtains the next 15-bit parcel for execution within a 60-bit instruction word. A parcel RNI is initiated when a sequence exit (SEQEXT) is present along with the condition of ADVF. This indicates that an additional parcel is available for execution within the current instruction word.

### PARCEL COUNTER

The parcel counter located on the GL module is a 2-bit counter that is incremented sequentially through counts of 0, 1, 2 and 3. The counts correspond to the four parcels of each central memory instruction word, and provide gating controls to selector U2. The counter is advanced during each RNI sequence to enable the entrance of the next 15-bit parcel into the instruction translation network. If a translated parcel is found to be part of a 30-bit instruction, the parcel counter will be incremented to ensure that K is not taken as the next instruction.

## DETAILED PAK DIAGRAM (CPU 3.13)

### F REGISTER, E REGISTER I2, F ADDER

#### F REGISTER

The F register and E register together serve as input feeders to the F adder. The F register also functions as the output register for results from the F adder. Input to F is from I2. The output of F feeds the F adder and the F register fanouts which distribute the results computed in the F adder.

#### I2 SELECTOR

The I2 selector provides 18-bit selection of data from I3 or the F adder to the F register. I3I2 controls selection of I3 through I2 to F. The absence of I3I2 allows the F adder to be selected through I2 to F. The generation of I3I2 from GC module F12 also generates ENABF from the same GC module. This allows I3 data to be selected through I2 and automatically gated into F.

#### F ADDER

The F adder consists of a high speed arithmetic logic unit (ALU) capable of performing both arithmetic and logical functions. Arithmetic logic operations are selected by the FAS 0-3, FAM signals. Group carry propagate (FP0003 - FP1619) and carry generate

(FG0003 - FG1619) signals from the F-ALU are sent to the first stage carry look-ahead control on FA module G12. The first stage carry look-ahead provides internal carry signals (FC0407 - FC1619) back to the F-ALU. The first stage carry look-ahead also provides group propagate (FP0015, FP0420) and end around carry (FEAC) signals to the second stage carry function control on FB module H13. The carry function control provides the end around internal carry signal (FC0003) back to the F-ALU, and distributes end around carry (FEAC) to the processor controls.

In its normal operation, the F adder performs a ones complement full add operation for: increment instructions (CPU 3.21), normal jump, return jump instructions (CPU 3.18, 3.19), compare/move address sequence calculations (CPU 3.33-3.37); also P register advancement, addition of RA to the absolute memory address for initial start, and full RNI operations. In addition, the F adder performs various functions for exponent manipulation or ones counter for FAD/FMD instructions (CPU 3.24, 3.25, 3.26).

#### E REGISTER

The E register and the F register together serve as input feeders to the F adder. The E register receives its input from I3. Its output feeds the F adder.

## DETAILED PAK DIAGRAM (CPU 3.12)

### I3, I3 COMPLEMENT CONTROL

#### I3 SELECTOR

The I3 selector provides 18-bit input selection to the I3 complement control. Inputs to I3 are received from the following circuits:

Normalize Network NN 0-5	CPU 2.8
I39 0-17	CPU 2.28
K 0-t4. U3 0-2	CPU 2.1
A Register 0-17	CPU 2.2
B Register 0-17	CPU 2.2
I0 Selector 0-17	CPU 2.3
X Register 0-17	CPU 2.2
X Register 48-57, 58	CPU 2.2

Selection through I3 is enabled by generation, at the GC modules, of the appropriate select command (e. g., SELBI3 selects B through I3 to the I3 complement control). If no selection is made, zeros are gated through I3.

#### I3 COMPLEMENT CONTROL

The I3 complement control allows the I3 output to be complemented before it is sent to I2 and the E register. COMI3 from the GC module allows I3 to be complemented.

## DETAILED PAK DIAGRAM (CPU 3.11)

### SHIFT NETWORK RANKS 3 & 4

#### SHIFT NETWORK RANK 3

The third rank of the shift network provides for right and left shifts of 4, 8 and 12. The SK register bits 2 and 3 determine whether the output from rank 2 (SN2 0-107) will be shifted by this third rank. RS determines the shift direction, left or right. The shift network third rank output feeds the fourth rank of the shift network.

#### SHIFT NETWORK RANK 4

The fourth rank of the shift network provides for right and left shifts of 1, 2 and 3. The SK register bits 0 and 1 determine whether the output from rank 3 (SN3 0-107) will be shifted by this fourth rank. RS determines the shift direction, left or right.

The fourth rank also provides the right shift capabilities for bits 108-113. Compare/move operations, using a maximum shift count of 54, require that the last characters from bit positions 48-53 are gated to bits 108-113. The SK register bits 1 and 2 determine whether the output from rank 1 (S2A 48-53) will be shifted to bits 108-113 by this fourth rank.

#### SHIFT NETWORK LOGIC LAYOUT

The F11 modules perform the right and left shifts for the upper 60 bits (48-107) from the C register. The lower 48 bits (0-47) from C can be right shifted only. This shifting occurs on the FL modules. An FL module also performs the right shift of bits 48-53 to rank 4 bits 108-113.

Left shifts are circular with the high order bits starting at 107, reentering at bit 48. Only the upper 60 bits may be left shifted (F11 modules). The entire 114 bits may be right shifted. Right shifts are end off with sign extension.

## DETAILED PAK DIAGRAM (CPU 3.10)

### SHIFT NETWORK RANKS 1 & 2 NORMALIZE NETWORK

#### SHIFT NETWORK RANK 1

The first rank of the shift network provides for right shifts of 64. The SK register bit 6 determines whether the C register output will be shifted by this first rank. The shift network first rank output feeds the second rank of the shift network.

#### SHIFT NETWORK RANK 2

The second rank of the shift network provides for right and left shifts of 16, 32 and 48. The SK register bits 4 and 5 determine whether the output from rank 1 (SNI 0-107) will be shifted by this second rank. RS determines the shift direction, left or right. The shift network second rank output feeds the third rank of the shift network.

#### HIGH/LOW SELECT CIRCUIT

The high/low select circuit allows selection of C register bits 0-47 or 48-95, plus bits 96-107, to the X register. This selection is required for use by double precision instructions and floating divide instructions.

#### NORMALIZE NETWORK

The normalize network receives its input from the 48-bit coefficient contained in the C register bits 48-95. The normalize network is a static network that forms at its output the 6-bit normalize count. This count is sent to the SK register via 19. The shift network, which is under control of SK, left shifts the coefficient the number of places specified by the normalize count, until the most significant 1 bit of the coefficient is in the bit 95 position.

Since the normalize network assumes a positive quantity, the circuits on the FJ modules compare each bit of the coefficient with the sign bit. This produces a 48-bit positive quantity that is sent to the normalize network where it is divided into three groups of 16 bits each. Each group generates a 4-bit count, giving the location of the highest order 1 bit in that group. The three groups of 4 bits (NNA 0-3, NNB 0-3, and NNC 0-3) are fed to the second stage of the normalize network where the 6-bit normalize count is formed.

The second stage of the normalize network also detects if a normalize count of zero is being formed, and automatically adds  $60_8$  to the count. This ensures a shift count of  $48_{10}$  on a 24 instruction with a coefficient equal to zero.

# DETAILED PAK DIAGRAM (CPU 3.9)

## SK COUNTER, I19, I9 SELECTORS

### I19 SELECTOR

The I19 selector provides input selection of U3 register bits 0-5 or SCRX bits 1-5 through I19 to the I9 selector.

### I9 SELECTOR

The I9 selector provides input selection to the SK counter. The following circuits are fed to the I9 selector:

1. Normalize count
2. F register
3. I19 selector

NOTE: These inputs are received in complement form because I9 always complements.

In addition, I9 can generate its own internal constants of  $60_8$  or  $74_8$  for iteration counts required by the FMD sequence.

Input and constant selections through I9 are controlled by a 2-bit selection code SLI90 and SLI91. The selection codes generated provide the following input or constant selections through I9:

<u>Selection Code</u>	<u>Input or Constant Selection</u>	
0 . COMT64	$60_8$	→ I9
0 . COMT64	$74_8$	→ I9
1	Normalize Count	→ I9
2	I19	→ I9
3 . FLI28 . FCOM	F register	→ I9
3 . FLI28 . FCOM	F register	→ I9
FLI28	1's	→ I9

The F register is gated to I19 when a select code of 3 is generated. However, F may be complemented through I9 if the F register sign (F17) is negative.

### SK REGISTER

The SK register serves a dual purpose. It acts as a register for storing the shift count, and as an iteration counter for the FMD sequence. The SKS1 and SKS2 signals control the functioning of the SK register. The shift or iteration count sent from I9 is stored in SK when a preset function is generated. During iterative steps of the FMD sequence, a decrement function code reduces the count by one for every clock pulse. When active, the SKEQ0 signal indicates that the iteration count has been decremented to zero.

### It5 SELECTOR

Selector I15 provides 60-bit input selection from the H register and 18-bit input selection from the F register and selector I1. Selector I15's outputs provide one of the 60-bit data inputs at selector I5 bit positions 48-107. I15's input selection is determined by selection control signals F46X, FEXI15, and ESI15. These control signals allow four input selections through It5:

<u>Input Select Control</u>	<u>Input Selection</u>
1. F46X . FEXI15 . ESI15	tl register <sub>0-59</sub> ~ I15 <sub>0-59</sub>
2. F46X . FEXI15	<div style="display: inline-block; vertical-align: middle;"> <div style="display: inline-block; vertical-align: middle;">{</div> <div style="display: inline-block; vertical-align: middle;"> F register<sub>0-10</sub> ~ I15<sub>48-58</sub>  C107 ~ I15<sub>59</sub> </div> </div>
3. F46X . ESI15	It selector <sub>0-17</sub> ~ I15 <sub>6-23</sub>
4. F46X . ESI15	F register <sub>0-17</sub> ~ I15 <sub>0-17</sub>

F46X allows selection of H register bits 0-59 to It5 bits 0-59 for compare/move operations. FEXI15 allows gating of the biased exponent and coefficient sign (C107) from F register bits 0-10 to I15 bits 48-59. FEXI15 is enabled during shift and floating instructions to pack the computed exponent with the result coefficient. ESI15 allows gating of the I1 selector output to It5 bits 6-23 for ECS instructions. When ESI15 is selected, zeros are gated to I15 bit positions 0-5. F46X . ESI15 allows gating of the F register bits 0-17 to I15 bits 0-17. This input selection is used by 7X increment, population count (47) and ECS instructions.

## DETAILED PAK DIAGRAM (CPU 3.8)

C, H REGISTERS;  
115, 15 SELECTORS;  
15 COMPLEMENT CONTROL

### C REGISTER

The C register is 114 bits in size. Bits 0-107 serve as one of the input feeders to the D adder. Bits 108-113 are used by compare/move operations to catch the last character shifted from bit positions 48-107. The C register also provides a general path for data distribution to the processor. Its outputs feed the following circuits:

- |                         |          |
|-------------------------|----------|
| 1. D adder              | CPU 3.5  |
| 2. R Register           | CPU 3.28 |
| 3. 130 Selector         | CPU 3.28 |
| 4. Shift Network Rank 1 | CPU 3.10 |
| 5. Normalize Network    | CPU 3.10 |

### 15 SELECTOR

The 15 selector provides input selection of data from various sources within the processor to the 15 complement control. Inputs to 15 are received from the following circuits:

- |                         |          |
|-------------------------|----------|
| 1. D Register           | CPU 3.15 |
| 2. C Register           | CPU 3.8  |
| 3. X Register           | CPU 3.2  |
| 4. 115 Selector         | CPU 3.8  |
| 5. 145 Selector         | CPU 3.31 |
| 6. Shift Network Rank 4 | CPU 3.11 |

In addition to the above input selections, 15 can generate its own internal constants of  $4_8$ ,  $6_8$  or zeros to the 15 complement control. Input or constant selection is determined by a 3-bit code generated from the 15 control (CPU 3.7). Decoding of the selection code bits at 15 allows the following input or constant selections to be made:

<u>Selection Code</u>	<u>Input or Constant Selection</u>
0	0's → $15_{0-113}$
1	D Register → $15_{0-107}$
2	C Register → $15_{0-113}$
3	$4_8$ 's → $15_{0-107}$
4	$\left\{ \begin{array}{l} 145_{0-6} \\ 115_{0-59} \end{array} \right. \rightarrow \begin{array}{l} 15_{0-6} \\ 15_{48-107} \end{array}$
5	SN4 → $15_{0-113}$
6	X Register → $15_{48-107}$
7	$6_8$ 's → $15_{0-107}$

### 15 COMPLEMENT CONTROL

The 15 complement control allows the 15 output to be complemented before it is sent to the C register. Complement selection is controlled by a 2-bit complement code (15C1, 15C2).

### 11 REGISTER

The H register acts as a 60-bit holding register for compare/move operations. The H register outputs feed the 115 selector located on the same JA modules.

# DETAILED PAK DIAGRAM (CPU 3.7)

## 15 INVERTER CONTROL

The 15 inverter rank controls the input to the C register. There are a number of varied inputs, not all of which are the same length as the 114-bit C register. For this reason, the 15 controls are broken into separate select networks for various areas of 15. Each select network provides a 3-bit code that will be translated to make the 15 input selection (CPU 3.8). Inputs to the select networks are from the instruction sequences. Outputs are defined in the following table:

<u>15 BITS</u>	<u>SELECT CODE</u>		
	<u>Bit 2</u>	<u>Bit 1</u>	<u>Bit 0</u>
15 105 - 107	15S436	15S236	15S136
15 96 - 104	15264	15164	15064
15 48 - 95	15285	15185	15085
15 45 - 47	15S418	15S218	15S118
15 9 - 44	15294	15104	15004
15 0 - 8	15208	15104	15004

**DETAILED PAK DIAGRAM (CPU 3.6)**  
**LARGE ADDER CARRY FUNCTION**

The FA, FB and HC modules shown on CPU 3.6 depict the large adder carry look-ahead and carry functions. Group carry propagate (PG0003 - PG4107) and carry generate (GG0003 - GG4107) signals are sent from the D-ALU bits 0-107 to the first stage carry look-ahead control. This control consists of seven FA modules divided into two groups. The first group provides internal carry signals (CN0003 - CN0047) back to the D-ALU bits 0-47, and provides second stage group carry propagate (P10015, P11631, P13247) and carry generate (G10015, G11631, G13247) to the second stage carry look-ahead control for bits 0-47. The second group of first stage carry look-ahead FA modules provides internal carry signals (CN4851 - CN4107) back to the D-ALU bits 48-107, and provides second stage group carry propagate (P14863, P16479, P18095, P19807) and carry generate (G10015, G16479, G18095, G19607) to the second stage carry look-ahead control for bits 48-107.

The second stage carry look-ahead controls located on FA module H12 and HC module H14 provide group carry propagate and carry generate signals from the second stage carry look-ahead back to the first stage carry look-ahead controls. It also provides the end around carry signals (C11Y047, CRY107) to the carry function control. This control, located on the FB module, provides the end around carry signals (LEAC, HEAC) to the first stage look-ahead, and distributes end around carry from bits 48-107 (LEAC) to the processor controls.

## DETAILED PAK DIAGRAM (CPU 3.5)

### D REGISTER, D ADDER, I14, I4

#### D REGISTER

The D register and C register together serve as input feeders to the D adder. The D register also functions as the output register for results from the D adder. I4 provides a 108-bit input to the D register from selector I14.

#### I14 SELECTOR

The I14 selector provides 108-bit selection of data from the D register or D adder to the I4 selector. I14S controls selection of the D register output through I14 to I4. The absence of I14S allows the D adder output to be gated through I14 to I4. The generation of I14S is controlled by FDI14 from the FAD/FMD sequence controls (CPU 3.26). FDI14 at HC module I114 generates I14S if CRY107 or  $\overline{44 + 45}$ .

#### I4 SELECTOR

The I4 selector provides 108-bit selection of data from I14 to the D register. Signals I40 and I41 control selection through I4. Four transfers are possible:

1. I14  $\rightarrow$  I4 (No shift)
2. I14  $\rightarrow$  I4 (Right shift one)
3. I14  $\rightarrow$  I4 (Left shift one)
4. 0's  $\rightarrow$  I4

The right and left shift capabilities are internal to I4. The right shift is end off, no sign extension; the left shift is not end around. The generation of I40 and I41 is controlled by the FAD/FMD sequence controls (CPU 3.26). I40 and I41 are also generated every common time 64 to gate 0's to the D register.

#### D ADDER

The D adder consists of a high speed arithmetic logic unit (ALU) capable of performing both arithmetic and logical functions. Arithmetic logic operations are selected by the DAS 0-3, DA-M signals. Group carry propagate (PG0003 - PG4107) and carry generate (GG0003 - GG4107) signals from the D-ALU are sent to the large adder first stage carry look-ahead control (CPU 3.6). The first stage carry look-ahead (FA modules H06, H11, H15, 106, H12, H13, H19) provides internal carry signals (CN0003 - CN4107) back to the D-ALU.

In its normal operation, the D adder performs a ones complement full add operation for: Boolean instructions (CPU 3.23), ECS instructions (CPU 3.27), integer sum/difference instructions and floating point instructions controlled by the FAD/FMD sequences (CPU 3.24, 3.25, 3.26).

The D adder also performs logical operations (inclusive OR, exclusive OR, logical AND) using 60-bit operands for Boolean instructions (CPU 3.23).

#### K1 REGISTER (SOURCE FIELD CM ADDRESS REGISTER)

The K1 register is used exclusively by compare/move instructions to specify the source field CM address.

#### K2 REGISTER (RESULT FIELD CM ADDRESS REGISTER)

The K2 register is used exclusively by compare/move instructions to specify the result or source field CM address.

#### P, RA, MA, FL LOGIC LAYOUT

The P, RA, MA and FL registers are contained on the FP modules. The CR9 register output provides an input path to P, RA, MA and FL during an exchange jump. A second input is provided to the P register from the F register. This path is used to store the updated contents of P back into the P register during a full RNI or branch, and to feed the address range test circuit also located on the FP module.

#### I0 SELECTOR

The I0 selector circuit provides input selection of the P, RA, MA, FL, K1 and K2 registers. The I0 selection code determines which of the six registers will be selected through I0. The I0 output feeds the data transmitters for storing P, RA, MA and FL during an exchange jump and provides an input path via I3 to the small adder.

#### RAE, FLE LOGIC LAYOUT

The RAE and FLE registers are contained on the FQ module. The CR9 register output provides an input path to RAE and FLE during an exchange jump.

#### I1 SELECTOR

The I1 selector circuit provides output selection of the RAE and FLE registers. The I1 output feeds the data transmitters for storing RAE and FLE during an exchange jump, and provides an input path via I15 and I5 to the large adder.

#### K1, K2 LOGIC LAYOUT

The K1 and K2 registers are contained on the JK module. The CR9 register output provides an input path to K1 and K2 via I49. CR9 is normally selected through I49 to gate the K1 and K2 portions of an instruction word into their respective registers. The second input path to K1 and K2 provides gating the updated K1 or K2 values from the F register back into the K1 or K2 registers during the address sequencing for a compare/move instruction.

#### EM, EE LOGIC LAYOUT

The EM and EE registers are contained on the FW module. The CR9 register output provides an input path to the EM register during an exchange jump. Inputs to the EE register consist of 3 error signals (AORI, INFOPR, INDF) generated within the CPU, and 3 error signals (DB+RDP, PARERR, FLGPAR) sent from the CMC.

The EM/EE registers feed the selector and error condition sense circuits. The selector circuit allows selection of the EM register to the data transmitters during an exchange jump, or of the EE register during an error exit sequence. The condition sense circuit compares each bit stored in the EE register with the exit mode bits in the EM register. When an error condition sets the respective EE register bit, and the exit mode bit in the EM register is also set, the ECONDS signal is generated to enable the return jump error exit sequence.

# DETAILED PAK DIAGRAM (CPU 3.3 & 3.4)

P, RA, MA, FL, RAE, FLE, EM/EE;  
K1, K2 REGISTERS;  
I0, I1, I49 SELECTORS

## P REGISTER (PROGRAM ADDRESS REGISTER)

The P register is an 18-bit register that contains the program address of a 60-bit instruction word currently being executed. The initial contents of P are provided by the exchange jump package and incremented by +1 for each program step. The contents of P are modified by the addition of RA (P + RA) to determine the central memory location for each instruction word.

## RA REGISTER (REFERENCE ADDRESS REGISTER)

The RA register contains a predetermined reference CM starting address for the current program in progress. RA is added to the address before each CM read or write reference, thus allowing multiprogramming and relocation of programs in central memory.

## MA REGISTER (MONITOR ADDRESS REGISTER)

The MA register contains the absolute starting address of an exchange package that is used when executing a central exchange jump instruction, or when honoring a MAN exchange request from PPS if the monitor flag (MF) is clear.

## FL REGISTER (FIELD LENGTH REGISTER)

The FL register is used to define the program field size. Before RA is added to an address for central memory, the address in the F register is checked against FL to determine if the upper limit of the program has been exceeded.

## RAE REGISTER (REFERENCE ADDRESS ECS REGISTER)

The RAE register contains a 21-bit relative starting address for ECS. The lower 6 bits of RAE are always zero.

## FLE REGISTER (FIELD LENGTH ECS REGISTER)

The 24-bit FLE register is used to define the program field size for ECS. The lower 6 bits of FLE are always zero.

The RAE and FLE registers serve the same purpose for ECS as do RA and FL for CM.

## EM/EE REGISTERS (EXIT MODE, ERROR EXIT REGISTERS)

The EM register contains the exit mode selections for a program in operation. The exit mode bits control CPU action if the corresponding error is detected. Six exit mode bits are provided as follows:

<u>Mode Selection Bit</u>	<u>Condition Sensed</u>
48	Address out of range
49	Operand out of range
50	Indefinite operand
57	Parity error in ECS flag register operation
58	CMC input error
59	CM data error

The EE register is set by the actual error conditions sensed by the CPU or sent to the CPU from CMC. The error condition signals which set the respective EE bits are as follows:

<u>Error Exit Bit</u>	<u>Condition</u>
AORI - 48	Address out of range
INFOPR - 49	Operand out of range (infinite operand)
INDF - 50	Indefinite operand
FLGPAP - 57	Parity error in ECS flag register operation
PARERR - 58	CMC input error
DB+RDP - 59	CM data error

## DETAILED PAK DIAGRAM (CPU 3.2)

### X, A, B REGISTERS

#### X REGISTERS

The X registers provide buffering between memory and the execution hardware. All 60-bit operands needed for instruction execution are obtained from the X registers, and all 60-bit results are returned to the X registers.

#### A REGISTERS

The A registers provide means for moving data between memory and the X registers, and comprise eight 18-bit address registers (A0 - A7). An address placed in an address register (A1 - A5) causes an immediate central memory reference to that address, and loads the operand from memory in the corresponding X register (X1 - X5). Placing an address in one of the two remaining address registers (A6 - A7) stores the word from the corresponding X6 or X7 register in the central memory address specified by A6 or A7.

The A0 register operates independently of X0 in that a change to the contents of A0 does not initiate a central memory reference.

#### B REGISTERS

The B registers consist of eight 18-bit indexing registers that have no connection to central memory. The B registers are manipulated by increment, pack and unpack instructions and can be used for control of program branching. B0 is maintained as a constant zero index.

#### X, A, B LOGIC LAYOUT

The X, A and B registers are contained on the FR modules. Each module consists of 4 MOS memory chips providing 16x4 bits of register storage. Only 8x4 bits are used on each module.

C, A and B register input selection is provided on each FR module. The X registers can receive input data from CR9 during exchange jump operations, or from the high/low C register select circuit. The A and B registers can receive input data from CR9 during exchange jump operations, or from the F register. The TLK (write) strobe is used to load the register, preventing a continuous read of the selected register.

#### X, A, B SELECTION

To read or write the contents of any one of the eight X, A or B registers, a 3-bit address is generated from the GK module. The GK module allows selection of the i, j or k portions of the instruction in U3 to be used for register addressing. For exchange jump operations, the GK module provides a sequence decode circuit to generate the required register addresses.

**DETAILED PAK DIAGRAM (CPU 3.1)**  
**U1, U2, RNI HOLD, U3, CONSTANT GENERATOR**

The U1 register, RNI hold register, U2 selector and U3 register are contained on the FU module.

U1, RNI hold and U2 disassemble each 60-bit memory word into four 15-bit parcels. These parcels are then translated sequentially to determine the instruction format of the memory word.

U1 holds the 60-bit memory word from CR9 during RNI initial start or full RNI operations. Parcels 0 and 1 from U1 are gated sequentially through U2 to U3, and parcels 2 and 3 are stored in the RNI hold register during execution of a full RNI for the next four parcels from memory.

The output of U2 is sent to U3 for instruction translation. If during translation, parcel 0 in U3 is found to be part of a 30-bit instruction, parcel 1 will be gated into I3 directly from U2. This forms the K portion (K bits 0 - 14) of the instruction.

The U3 register feeds the function decode logic contained on the FK module, and provides instruction designator fanouts for the f, m, l, j and k bits of the instruction word.

An RNI exit to common time advances the parcel counter which gates the next parcel into U2. The next parcel RNI gates U2 to U3, translating the next instruction.

**CONSTANT GENERATOR**

The constant generator circuit located on the FV module generates constant values required by the FAD and FMD sequences. The constant generator is capable of generating constant values of 1, 57<sub>8</sub> and 60<sub>8</sub>. The constant value is sent via I39 to I3.

DETAILED PAK DIAGRAM (CPU 3.0)  
INPUT DATA AND CONTROL REGISTERS

The CR9 and input control registers located on the KR modules are the data input and control signal catching registers for the CPU.

CR9 REGISTER

The CR9 register receives 60-bit input operands or instructions from CMC. Parity for each 60-bit word is checked on the KC module with the input parity bit IDTPAR. Should an input parity error be detected, INPARE sets the input parity error FF during the accept sequence.

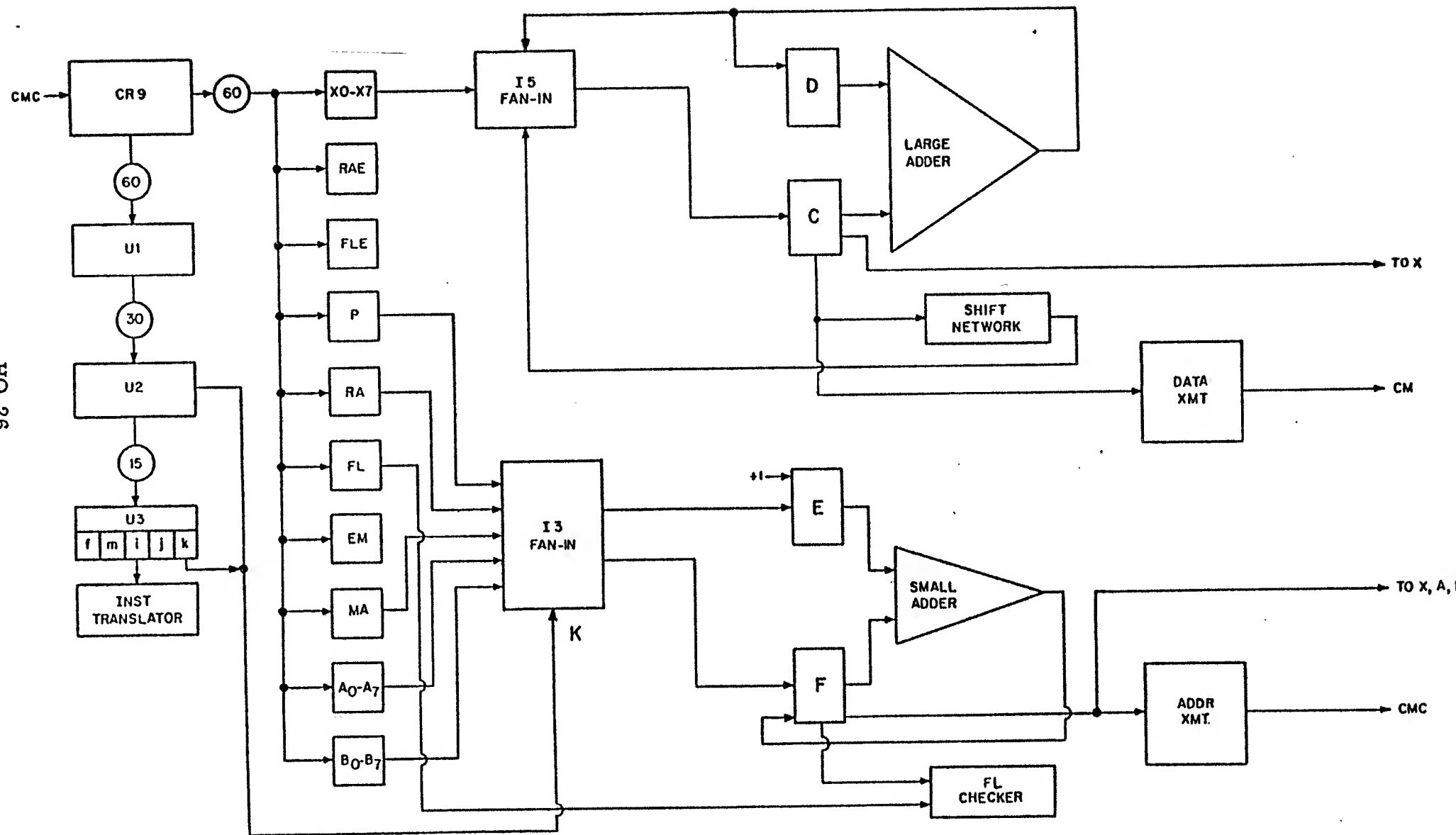
CONTROL REGISTERS

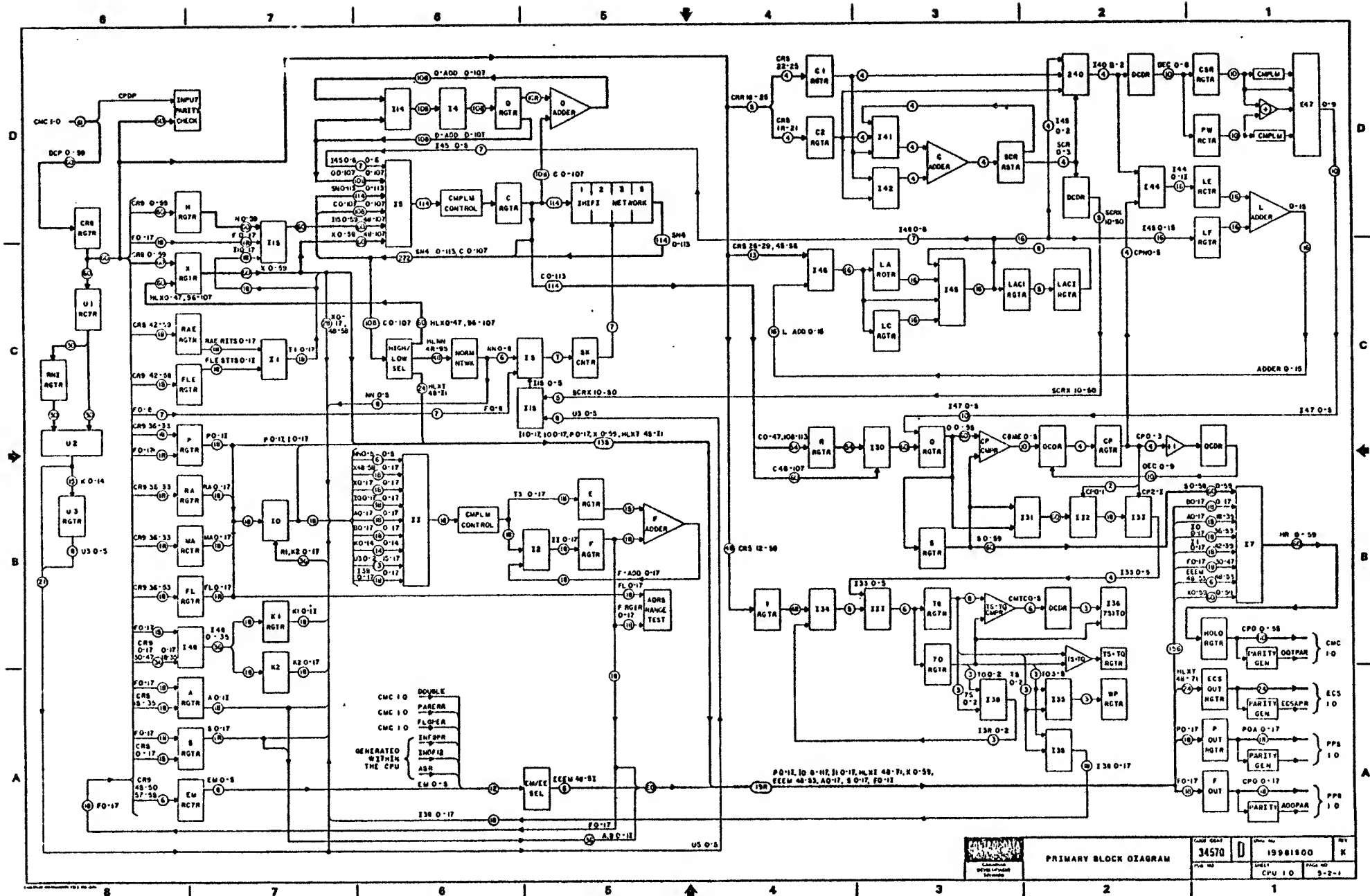
The input control registers receive control signals and parity information from CMC, PPS and the ECS Coupler. The control signals received are defined as follows:

REQEXJ	<u>Request Exchange:</u> CMC requests the start of an exchange jump sequence.
IDTPAR	<u>Input Data Parity:</u> Data parity for 60 data bits sent to CR9.
DOUBLE	<u>Double Data Error:</u> Double data error detected by SECDED.
DATRDY	<u>Data Ready:</u> CMC sends data ready 50 ns ahead of output data.
ACCEPT	<u>Accept:</u> CMC sends accept when a request for CM access has been accepted and a memory cycle is initiated.
PARERR	<u>Parity Error:</u> A parity error detected in the input data or address information. This signal is sent from CMC at the same time as accept; however, a memory reference is inhibited.

BKPTFL	<u>Breakpoint Flag:</u> CMC sends the breakpoint flag when a breakpoint condition is met during a CPU to CM access.
MONFLO	<u>Monitor Flag:</u> This signal from CMC indicates the state of the monitor flag FF.
NCEJSW	<u>MEJ/CEJ Switch:</u> This signal from the PPS indicates whether the MEJ/CEJ switch from the dead start panel is in the ENABLE or DISABLE position.
NMASCL	<u>Processor Master Clear</u>
ENDTRF	<u>ECS End Transfer:</u> ECS coupler sends end transfer when an ECS transfer is completed normally.
ERRABT	<u>ECS Error Abort End Transfer:</u> ECS coupler sends error abort when an error condition has been detected.
ADDPAR	<u>Zero Address Parity:</u> This signal causes the parity bit, on addresses transmitted to the CMC, to be a constant zero.
OUTPAR	<u>Zero Data Parity:</u> This signal causes the parity bit, on data transmitted to the CMC, to be a constant zero.
FLGPER	<u>Flag Register Operator Parity Error:</u> Indicates detection by the CMC of a parity error on an ECS instruction affecting the flag register.
ECSACP	<u>ECS Accept:</u> indicates readiness of the ECS coupler to process the ECS starting parameters from the CPU.

## CYBER 173 CPU BLOCK DIAGRAM





#### COMPARE/MOVE DATA SECTION

The compare/move data section consists of:

- H, R, Q, S and T registers
- Unequal character position register (CP)
- Word comparison circuits
- Character comparison circuits
- Collate character comparison circuits
- TS, TQ and WP registers

#### COMPARE/MOVE CONTROL SECTION

The compare/move control section consists of:

- C1 and C2 offset registers
- 4-bit C adder
- Shift count register (SCR)
- Character select (CSR) and partial write (PW) registers
- Field length registers LA, LC, LAC1 and LAC2
- L adder and LE and LF feeder registers.

#### OUTPUT SECTION

The processor output section consists of the hold register (HR), ECS output register, P output register and F output register. Data and control signals are sent from the output section to CMC, PPS and the ECS coupler.

## PRIMARY BLOCK DIAGRAM (CPU 1.0)

### INPUT REGISTER

The CR9 register receives all input data in the form of instructions or operands.

### INSTRUCTION CONTROL REGISTERS

The Ut, RNI, U2 and U3 registers form the portion of the processor that handles instructions. All instructions from CM are sent to CR9. From CR9, they continue to U1 where they are disassembled. From U1, instructions are sent to the RNI register, or via U2 to the U3 register. At U3, instructions are decoded and the appropriate control sequences are enabled to execute the instruction.

### OPERATING AND CONTROL REGISTERS

The control registers include:

Program Address register	(P)
Reference Address register	(RA)
Monitor Address register	(MA)
Field Length register	(FL)
Reference Address for ECS register	(RAE)
Field Length for ECS register	(FLE)
Source Field Address register for compare/move	(K1)
Destination Field Address register for compare/move	(K2)
Exit Mode register	(EM)

These registers are used in conjunction with instruction controls during the execution of instructions.

The operating registers hold operands used during the execution of instructions. There are 24 operating registers divided into three groups of eight registers each: the address registers (A), the index registers (B), and the operand registers (X).

### LARGE ARITHMETIC SECTION

The large arithmetic section consists of:

- 108-bit D adder
- 114-bit C register
- 108-bit D register
- Input selectors to the D register I14 and I4
- Input selectors to the C register I1, I15 and I5
- Shift network
- High/low select circuit
- Normalize network
- Input selectors to the iteration and shift counter I18 and I9
- SK shift and iteration counter.

The large arithmetic section is used during the execution of instructions using 60-bit operands. It includes all multiply, divide, logical, add, shift, compare/move, and ECS instructions.

### SMALL ARITHMETIC SECTION

The small arithmetic section consists of:

- 18-bit F adder
- 18-bit E and F registers
- Input selectors to the F register I0, I3 and I2
- Input selectors to the E register I0, I3
- Address range test
- F register test circuits.

The small arithmetic section handles instructions using 18-bit operands; these include increment and jump instructions. This section also handles exponent manipulation for floating point instructions and compare/move address calculations.

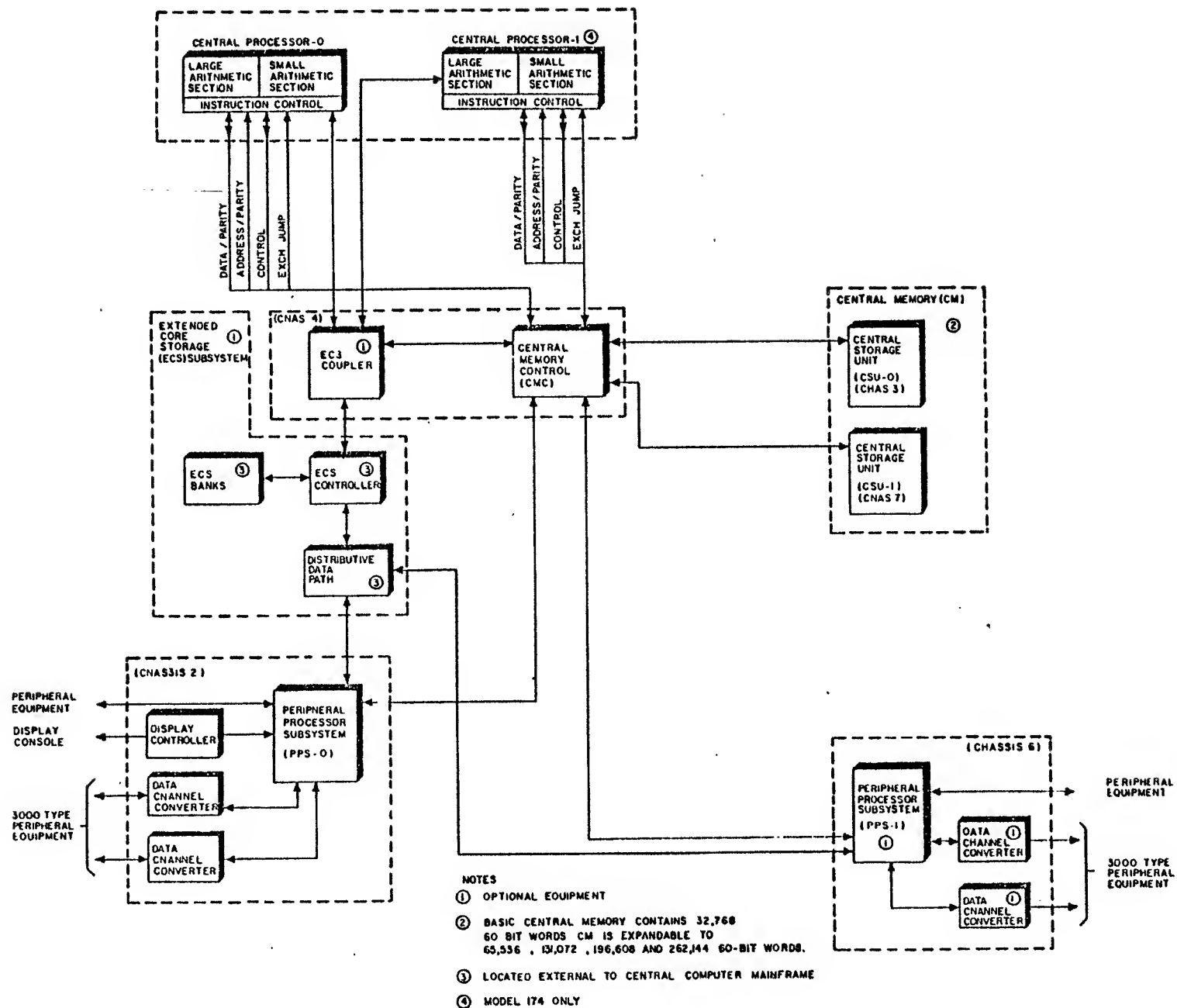
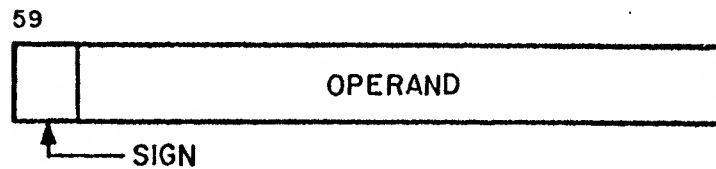
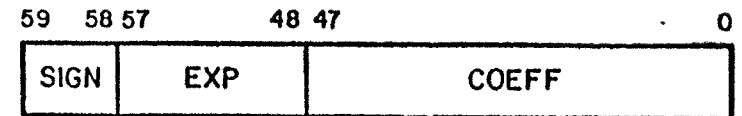
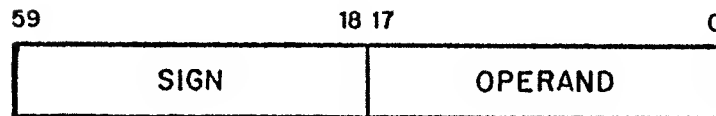
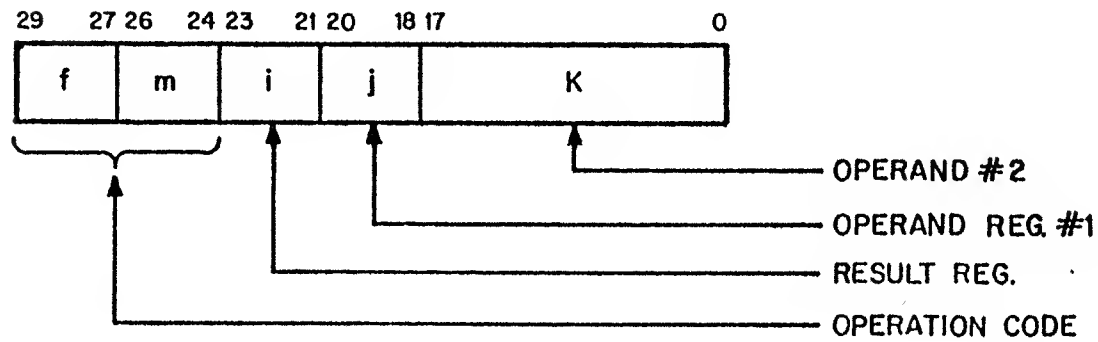
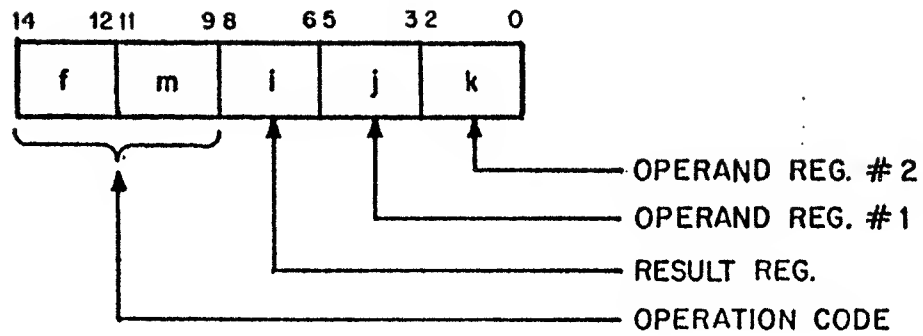
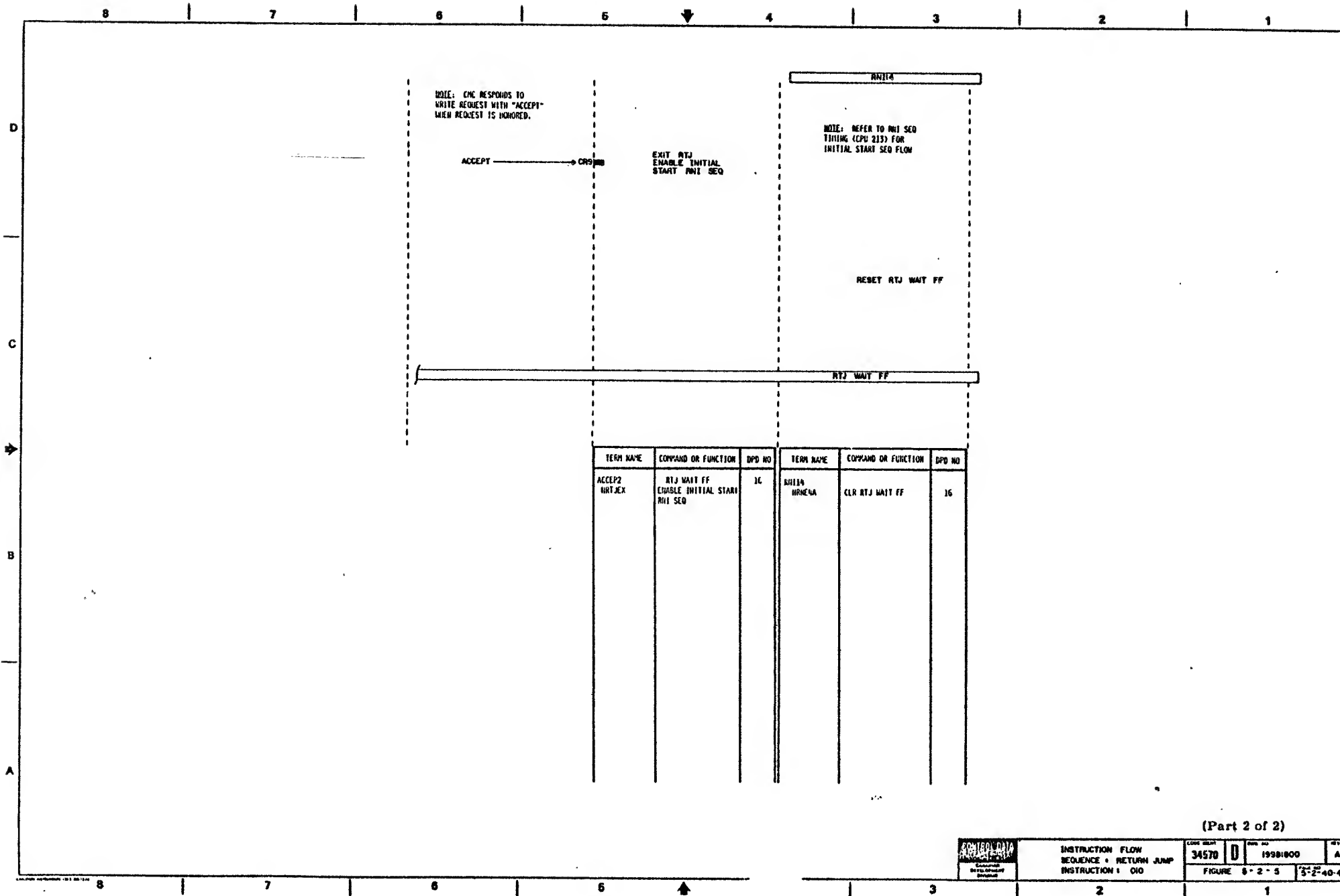


Figure 5-1-1. System Block Diagram

171-174

## CPU INST. &amp; OPERAND FORMAT



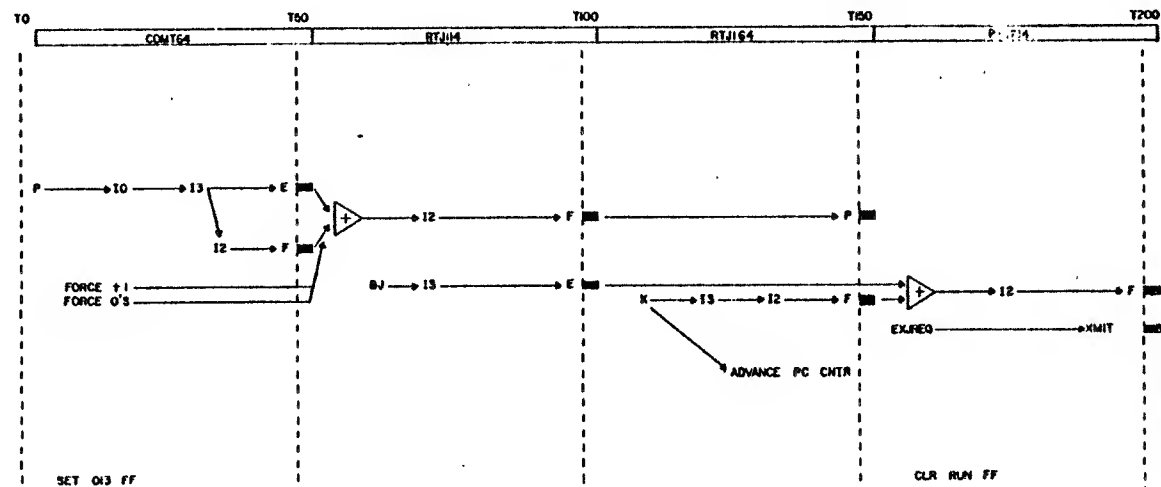


(Part 2 of 2)



INSTRUCTION FLOW  
SEQUENCE • RETURN JUMP  
INSTRUCTION 1: 010

CODE SHEET	34570	REV. NO.	19981800	REV.	A
FIGURE 5-2-5		3-2-40-4			



TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO
NR100A	P → 10	15	NR100B	[PC-1]	19	RTJ114		19	RTJ214		19
SL100		3	SEL RF		13	NR114A		19	NR114A		19
ENABE	ENABLE E NGIR	13	SFOO	+1 → F		1312	13 → 12	13	ENABF	ENABLE F NGIR	13
EN1013		15	RTJ114		19	ENABF	ENABLE F NGIR	13	RTJ214	[013 FF]	19
SL1013	10 → 03	12	NR114A		19	RTJ114		19	ELRR114	CLR RUN FF	19
NR100A		15	ENABE	ENABLE E NGIR	13	NR114A		19	RTJ214		19
1312	13 → 12	13	ENABF	ENABLE F NGIR	13	ENABP	ENABLE P NGIR	3	EXJREQ	EXJREQ → XMIT	19
EN OF	ENABLE F NGIR	13	NRJ100	SELECT RJ NGIR	2	NR114A		19			
013	SET 013 FF	19	SEL RJ		9	SELK13	K → 13	12			
			NR114A	B → 13	12	ADVPC2	ADVANCE PC CNTR	14			
			SELB13								
			NR100A	[PC-1]	19						
			ELRF	CLR F NGIR	13						
			ELRF00	0 → F0	13						

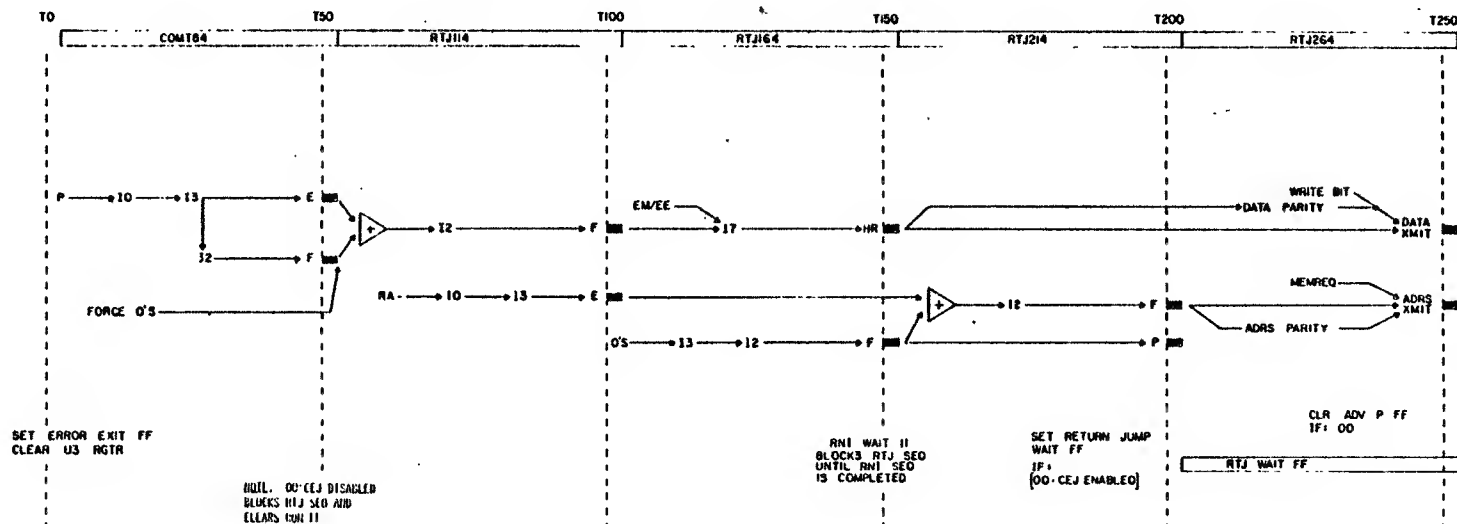
NOTE: O/C RESPONSE TO  
"RJ114" WITH "NRJ114"  
WHICH INITIATES EXCHANGE  
EXCHANGE SW (2.19)  
1. N/A CONTAINS EXCHANGE  
PACKAGE ADDRESS



INSTRUCTION FLOW  
SEQUENCE: RETURN JUMP  
INSTRUCTION: 013

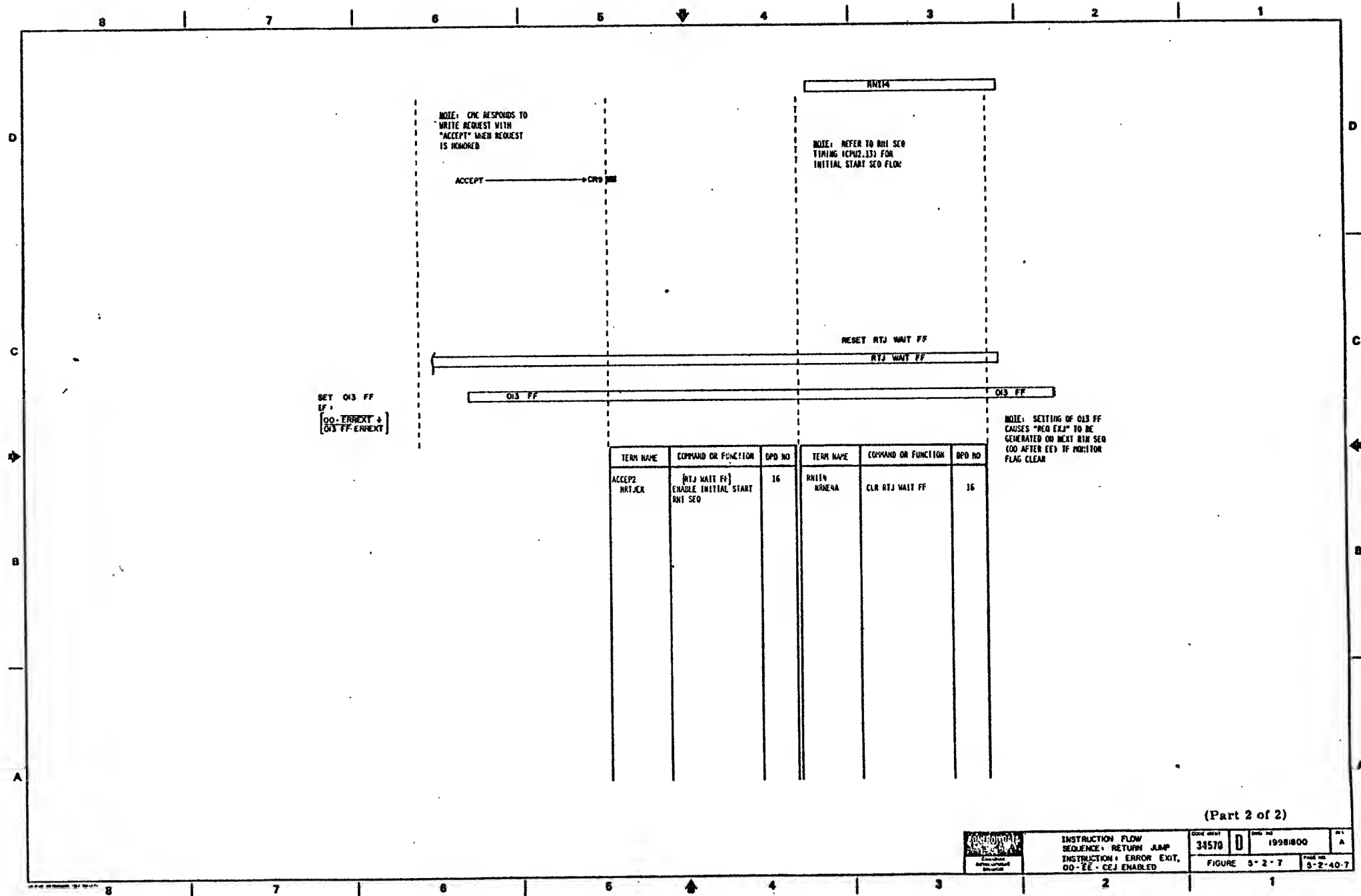
DATE: 1998/00  
FIGURE 5-2-6  
5-2-40-5

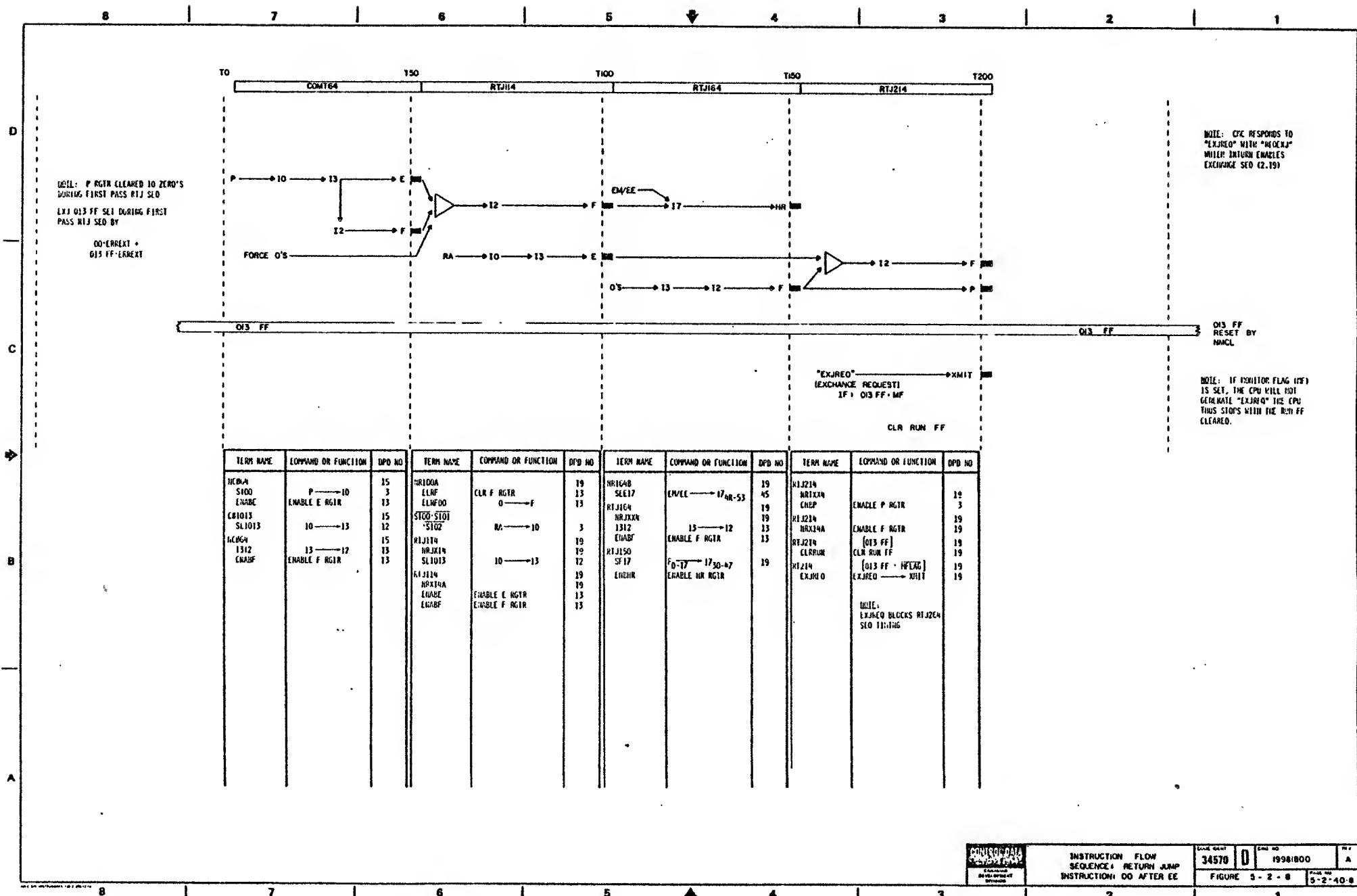
RULES: AN "11" VALUE  
 OF C'S, CARRY:  
 11: ILLGAL INSTRUCTION  
 013: PC-7  
 013: CEJ DISABLED  
 014: DIS + 01C + 017  
 1CS: ILLGAL  
 CFW: ILLGAL  
 29: 50: BIT ILLGAL  
 50: 1:11: CONDITION SENSED  
 AND:  
 HIGH INTR  
 IN INTR  
 AND'S PARITY  
 DATA PARITY  
 DATA ERROR  
 41: BREAKPOINT SENSED  
 ERROR EXIT FF (2:1C) IS SET  
 AT SEQUENCE EXIT. ERROR  
 EXIT FF CARRY IS "CARRY"  
 TO CLEAR US INSTRUCTION OR  
 BREAKPOINT.



TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO
REG164	P → 10	15	REG164	CLR F RGTR	19	REG164	ENR164	19	REG164	ENR164	19	REG164	ENR164	19
SL1013	ENABLE F RGTR	13	SL1013	CLR F RGTR	13	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19
SL1013	10 → 13	12	SL1013	RA → 10	3	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19
SL1013	13 → 12	15	SL1013	10 → 13	19	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19
SL1013	ENABLE F RGTR	13	SL1013	10 → 13	12	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19
SL1013	(00: CEJ DISABLED)	19	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19
SL1013	CLR F RGTR	13	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19	SL1013	ENR164	19

(Part 1 of 2)





INSTRUCTION FLOW  
SEQUENCE: RETURN JUMP  
INSTRUCTION: DO AFTER EE

DATE SENT 34570  
DATE NO 1998/800  
FIGURE 5-2-8  
5-2-40-8

# DETAILED PAK DIAGRAM (CPU 3.20)

## EXCHANGE SEQUENCE

An exchange jump can be issued from either the PPU or the CPU. An exchange jump interrupts the processor and causes it to exchange a 16-word package in central memory with the CPU registers. The package which the CPU receives contains all the initial operating parameters, and the package received by memory contains all the present operating parameters.

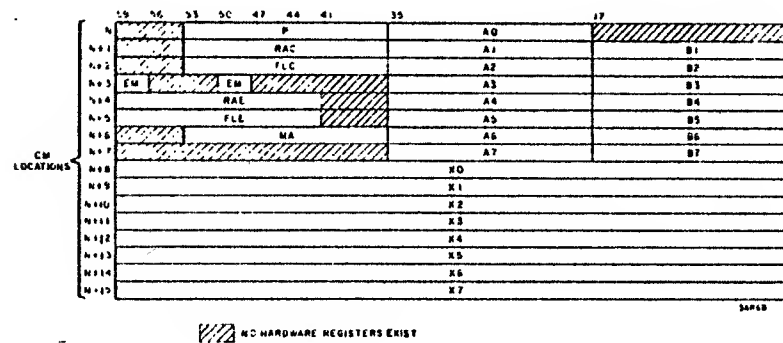
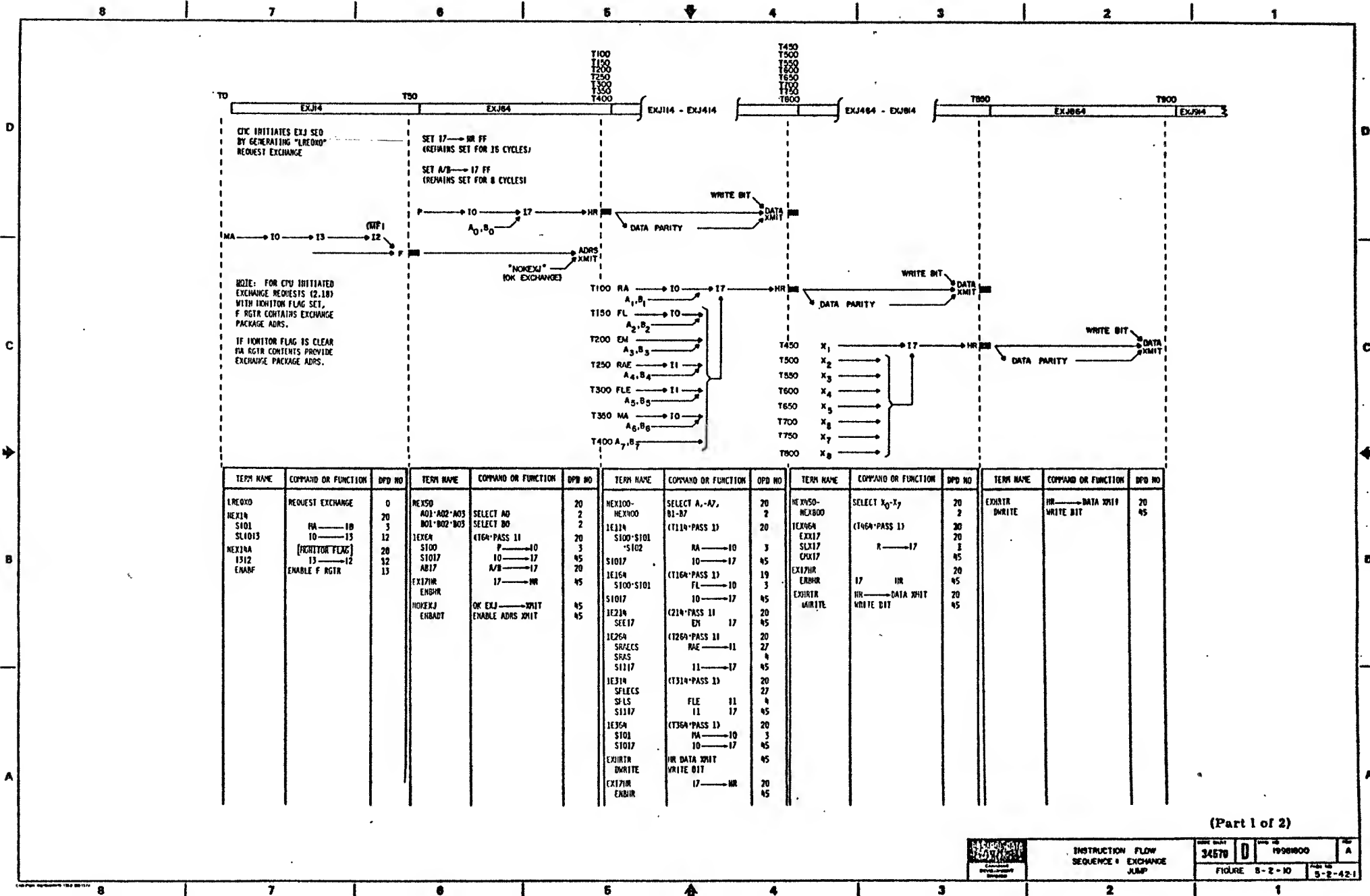


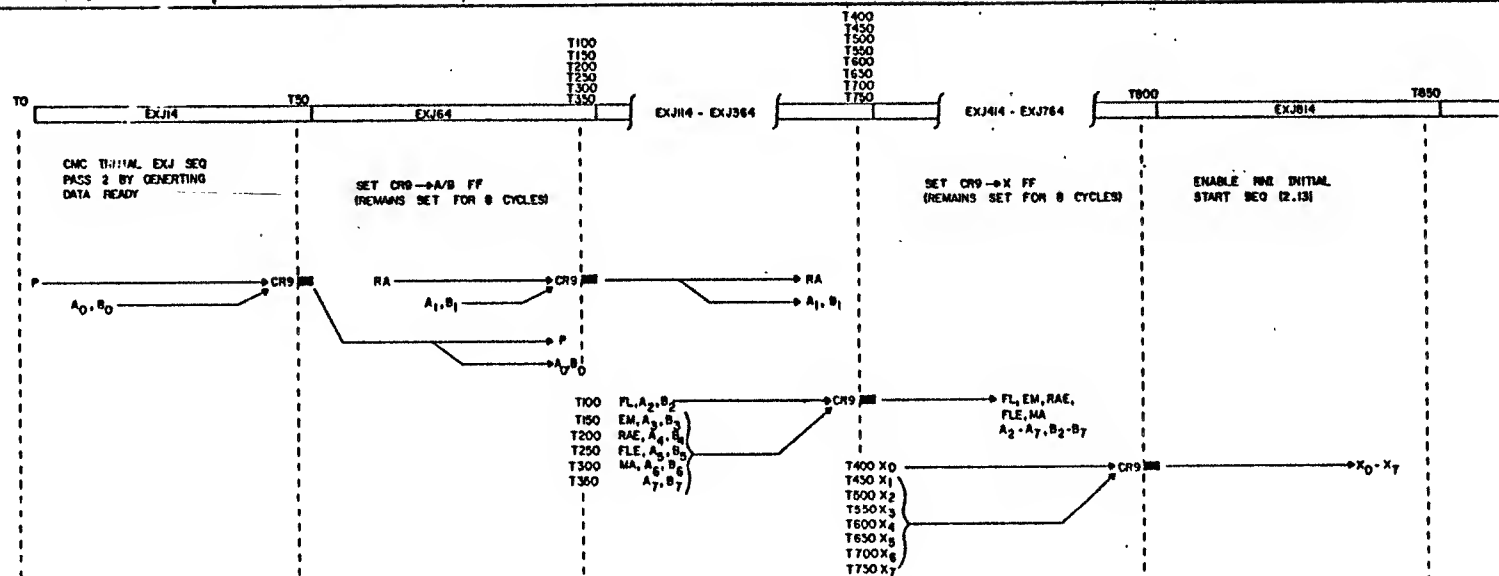
Figure 5-2-9. Exchange Package

The exchange sequence generates the necessary control signals to implement the exchange of data between the CPU and CMC. It also provides the internal controls to: enter the contents of the exchange package, interrupt the program currently being executed, and exchange the operating registers and control parameters with those of another program without information loss.

Whether the exchange request was first initiated by the CPU in response to a 0t3, 00/error exit instruction, or from the PPU, the exchange sequence is initiated by REQEXJ from CMC. An OK exchange signal (NOEXJ) is returned to CMC when the parcel count = 0 at instruction exit or the CPU is stopped. These conditions ensure that all instructions of the last 60-bit word have been executed before initiating the exchange.

At the CPU, the exchange takes 1.4 usec to complete.





TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO
DABY EXJ14	DATA READY	20	NEX50 A01-A02-A03 B01-B02-B03	SELECT A0 SELECT B0 (16A-PASS2)	20 2 2	NEX100- NEX350	SELECT A, -M6 B, -M6	20 2	NEX400- NEX750	SELECT A7, B7, X0-X6	20 2 2	NEX800	SELECT X7	20
			ZEX6A CR3P ENBP	CR9 -> P 36-53	20 3	ZEX11A ENBRA	(1314-PASS 2) CR936-53 -> RA	20 3	NCR9K	(16A-PASS 2) CR9 -> X FF	20 2	ZEX81A TNSRT	(181A-PASS 2) RNT INITIAL START	20 14
			NCR9AB NCR9A NCR9B ENABA ENABD	CR9 -> A/B FF  ENABLE A NGTR ENABLE B NGTR	20  2 2	ZEX16A ENBFL ENABEN FLXEE	(176A-PASS 2) CR936-53 -> FL (1214-PASS 2) CR936-53 -> EM 57-59	20 3 20 4	CR9X NENBX ENABX	ENABLE X NGTR	2 2			
						ZEX26A ENRAS	(126A-PASS 2) CR942-59 -> RAE	20 4						
						ZEX31A ENFLS	(1314-PASS 2) CR942-59 -> FLE	20 4						
						ZEX36A ENBPA	CR936-53 -> PA	20 3						

(Part 2 of 2)

# DETAILED PAK DIAGRAM (CPU 3, 21)

## INCREMENT SEQUENCE

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed point operands for increment instructions 50-77, and controls the formation of 60-bit ones complement sum and difference values for integer instructions 36 and 37.

### INCREMENT INSTRUCTIONS

SET A: 50ijk	Set Ai to (Aj) + K
51ijk	Set Ai to (Bj) + K
52ijk	Set Ai to (Xj) + K
53ijk	Set Ai to (Xj) + (Bk)
54ijk	Set Ai to (Aj) + (Bk)
55ijk	Set Ai to (Aj) - (Bk)
56ijk	Set Ai to (Bj) + (Bk)
57ijk	Set Ai to (Bj) - (Bk)

The 5x instructions perform a ones complement addition or subtraction of 18-bit operands. The 18-bit result is stored in address register Ai. Overflow is ignored, but an address range fault may result from overflow.

The first operand from the Aj, Bj or Xj register is selected at common time 64 and sent to the E register. For an operand selected from Xj, only the truncated lower 18 bits of the 60-bit word are sent to E.

The second operand selected at INC114 time from the Bk register or the K portion of the instruction itself (K = 18-bit signed constant) is sent to the F register. Selection of K causes the ADVPC2 signal to be generated (CPU 3-12) which, in turn, advances the parcel counter. The parcel counter will thus point to the next 15-bit instruction.

Addition or subtraction (by complement addition) of the two operands is performed through the F adder at INC164 time. The results are sent to the F register, and from F to the selected Ai register. If Ai  $\neq$  0, a range test is performed on the quantity in the F register at INC214 time to determine if the program range limits have been exceeded.

A reference is made to central memory using the newly created absolute address plus the reference address from iIA. The type of reference made is a function of the i designator value.

i = 0	No Memory Reference
i = 1-5	Read from Memory to Xi
i = 6, 7	Write into Memory from Xi

If the range test detected an address out of range condition, the following occurs independently of the exit mode selection.

If i = 1-5:	Xi is loaded with all zeros from CR9 and the contents of memory location Ai are unchanged.
If i = 6 or 7:	Xi retains its original contents and the contents of memory location Ai are unchanged.

SET B: 60ijk	Set Bi to (Aj) + K
61ijk	Set Bi to (Bj) + K
62ijk	Set Bi to (Xj) + K
63ijk	Set Bi to (Xj) + (Bk)
64ijk	Set Bi to (Aj) + (Bk)
65ijk	Set Bi to (Aj) - (Bk)
66ijk	Set Bi to (Bj) + (Bk)
67ijk	Set Bi to (Bj) - (Bk)

The 6x instructions perform a ones complement addition or subtraction of 16-bit operands. The 18-bit result is stored in increment register Bi. An overflow condition is ignored.

Operands for the 6x instructions are obtained in the same manner as described for the 5x instructions. A memory reference is not performed.

SET X:	70ijk	Set Xi to (Aj) + K
	71ijk	Set Xi to (Bj) + K
	72ijk	Set Xi to (Xj) + K
	73ijk	Set Xi to (Xj) + (Bk)
	74ijk	Set Xi to (Aj) + (Bk)
	75ijk	Set Xi to (Aj) - (Bk)
	76ijk	Set Xi to (Bj) + (Bk)
	77ijk	Set Xi to (Bj) - (Bk)

The 7x instructions perform a ones complement addition or subtraction of 18-bit operands. The 18-bit result is stored in the lower 18 bits of operand register Xi. The sign of the result is extended to the upper 42 bits of operand register Xi. An overflow condition is ignored.

Operands for the 7x instructions are obtained in the same manner as described for the 5x instructions. A memory reference is not performed. Sign extension is performed by the complement I5 control.

#### INTEGER SUM - DIFFERENCE

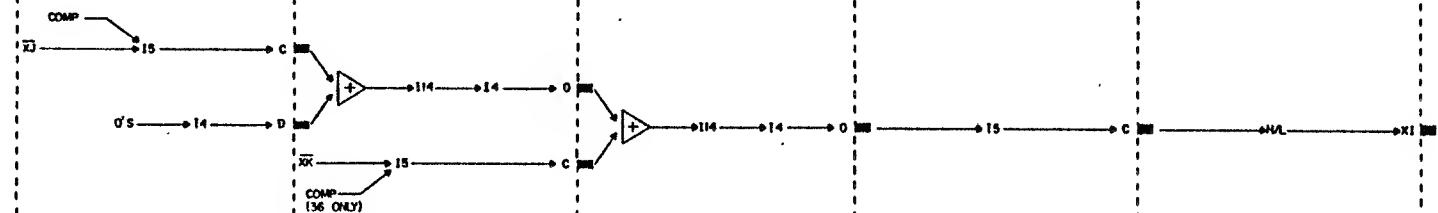
36ijk	Integer Sum of (Xj) and (Xk) to Xi
37ijk	Integer Difference of (Xj) and (Xk) to Xi

The 36 instruction forms a 60-bit ones complement sum of the quantities from operand registers Xj and Xk. Xj and Xk operands are considered as signed integers. Xj is sent to the D register, and Xk is sent to the C register. Integer addition is performed in the D adder, and the 60-bit result is stored in the Xi register. Overflow conditions are ignored.

The 37 instruction forms a 60-bit ones complement difference of the quantities from operand register Xj (minuend) and Xk (subtrahend). The 37 instruction allows the Xk complement to be sent to the C register; thus Xk is subtracted from Xj by complement addition through the D adder.

TO T30 T100 T150 T200 T250

COMT64 INC114 INC164 INC214 COMT14

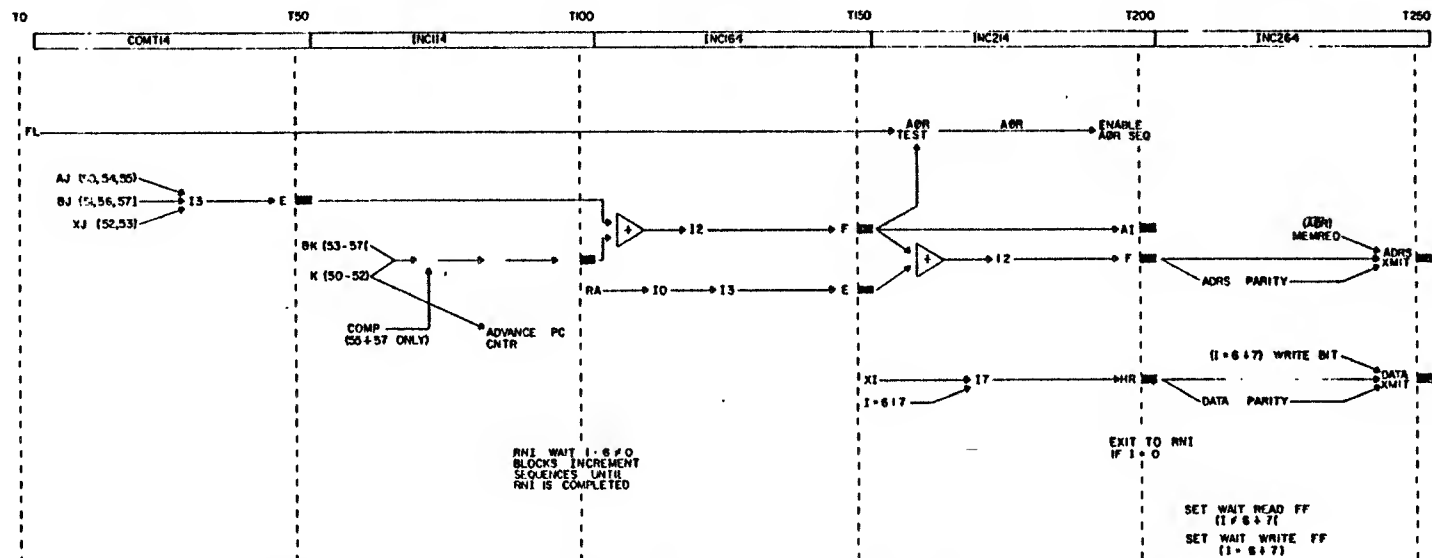


TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO
CM150		15	ITRS	D ADD → 114	5	ITRS	D ADD → 114	5	INC214	(F3637)	21	CM100		15
SELXJ	SELECT XJ RGR	2	114-114	114 → 114	5	114-114	114 → 114	5	NI214R		21	SELXI	SELECT XI RGR	2
HEP64		7	NI114R	ENABLE D RGR	21	NI114R	ENABLE D RGR	21	IS085	048-107 1548-107	7	COENK	ENABLE WRITE STROBE X RGR	2
15185	X0-59 → 1548-107	7	CHABD		5	NI114R	ENABLE D RGR	5	IS084		7			
15285		7	NI100	SELECT XK RGR	3	NI114R		21	155136	(1554-1552-1551)	7	CM100		15
15164		7	SNK		5	15185	X0-59 → 1548-107	7	NI114R	ENABLE C RGR	8	ILSL	SELECT HIGHER	10
15264		7	SELXK		5	15285		7	NI214R		21			
15236	(1554-1552-1551)	7	NI114		21	155236	(1554-1552-1551)	7	NI114R	CK11 TO COMMON TIME, RNI	21			
15436		7	15185		7	155436		7	NI114R		21			
CM150		15	15285		7	INC114	(F3636)	21	NI114R		21			
CHABD	ENABLE C RGR	8	155236		7	15C815	COMP 15	21	NI114R		21			
HEP64		15	155436		7	NI114R	ENABLE C RGR	8	NI114R		21			
114-114	OS → 114	5	INC114		21									
HEP50		15	15C815		21									
ENABD	ENABLE D RGR	5	NI114R	ENABLE C RGR	8									



INSTRUCTION FLOW  
SEQUENCE 1 INCR  
INSTRUCTION 1 36, 37

LOGO CODE 34570  
PAGE NO 19961800  
FIGURE 5-2-11  
PAGE NO 5-2-44-2



TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO	TERM NAME	COMMAND OR FUNCTION	DPD NO
CM50	SELECT XJ RGR	15	IN100	SELECT EX RGR	21	ST00 ST01	RA → 10	3	IN100A	ENABLE F RGR	13	IN100A	ENABLE F RGR	13
SELXJ	SELECT XJ RGR	2	SELBK	SELECT BK RGR	21	ST02			ENADF	ENABLE F RGR	13	IN100B	ENABLE F RGR	13
SELBJ	SELECT BJ RGR	2	IN100C	[53-57]	21	IN100A	10 → 13	12	IN100C	[1 + 0]	21	IN100B	ENABLE F RGR	13
CM150	SELECT AJ RGR	15	SEL13	E → 13	12	SL1013		13	ENADNR	ENABLE F RGR	13	IN100D	ENABLE F RGR	13
SELAN	SELECT AJ RGR	15	IN100D	[50-52]	21	ENADE	ENABLE E RGR	13	IN100D	SELECT A1 RGR	21	IN100D	ENABLE F RGR	13
CM150	SELECT XJ RGR	15	SEL13	R → 13	12	IN100A	ENABLE F RGR	13	IN100D	F → A	2	IN100D	ENABLE F RGR	13
SELXJ	SELECT XJ RGR	15	ADVPC2	ADVANCE PC COUNTER	12	IN100A	ENABLE F RGR	13	IN100D	ENABLE F RGR	13	IN100D	ENABLE F RGR	13
CM150	SELECT XJ RGR	15	IN100A	[53-57]	21	IN100A	ENABLE F RGR	13	IN100D	ENABLE F RGR	13	IN100D	ENABLE F RGR	13
CM150	SELECT XJ RGR	15	SC013	COMP 13	12	IN100A	ENABLE F RGR	13	IN100D	ENABLE F RGR	13	IN100D	ENABLE F RGR	13
CM150	SELECT XJ RGR	15	IN100A	13 → 12	21	IN100A	ENABLE F RGR	13	IN100D	ENABLE F RGR	13	IN100D	ENABLE F RGR	13
CM150	SELECT XJ RGR	15	IN100A	ENABLE F RGR	13	IN100A	ENABLE F RGR	13	IN100D	ENABLE F RGR	13	IN100D	ENABLE F RGR	13

(Part 1 of 2)

